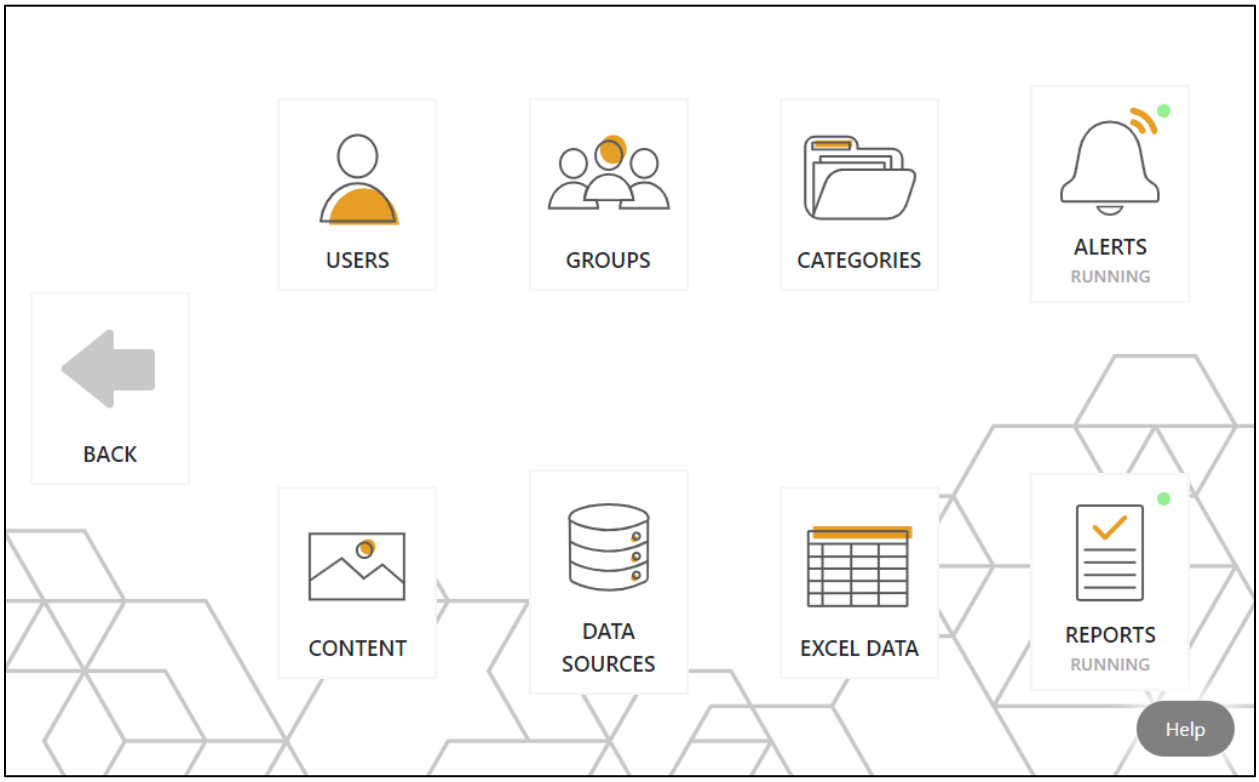
iDashboards

PRODUCT MANUAL



iDashboards Administrator's Manual

Version 11.3

iDashboards Administrator's Manual

Version 11.3

Support information:

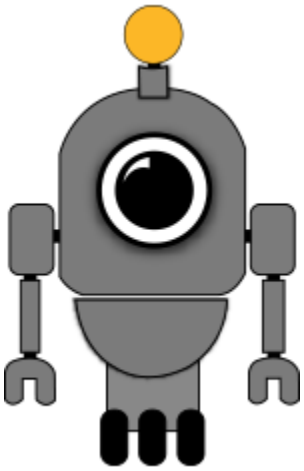
iDashboards
900 Tower Drive, 4th Floor
Troy, MI 48098

Phone: (248) 528-7160
Fax: (248) 828-2770

Email: support@iDashboards.com
Web site: http://www.iDashboards.com

1. Table of Contents

2. OSKAR



OSKAR, the Online Support & Knowledge Acquisition Repository, is the preferred support resource for iDashboards' customers, partners and prospects. The OSKAR Support Portal can be used to submit, review and update support tickets.

https://oskar.idashboards.com/

Those who have an active, support and maintenance contract with iDashboards also have access to the following content in our User Community:

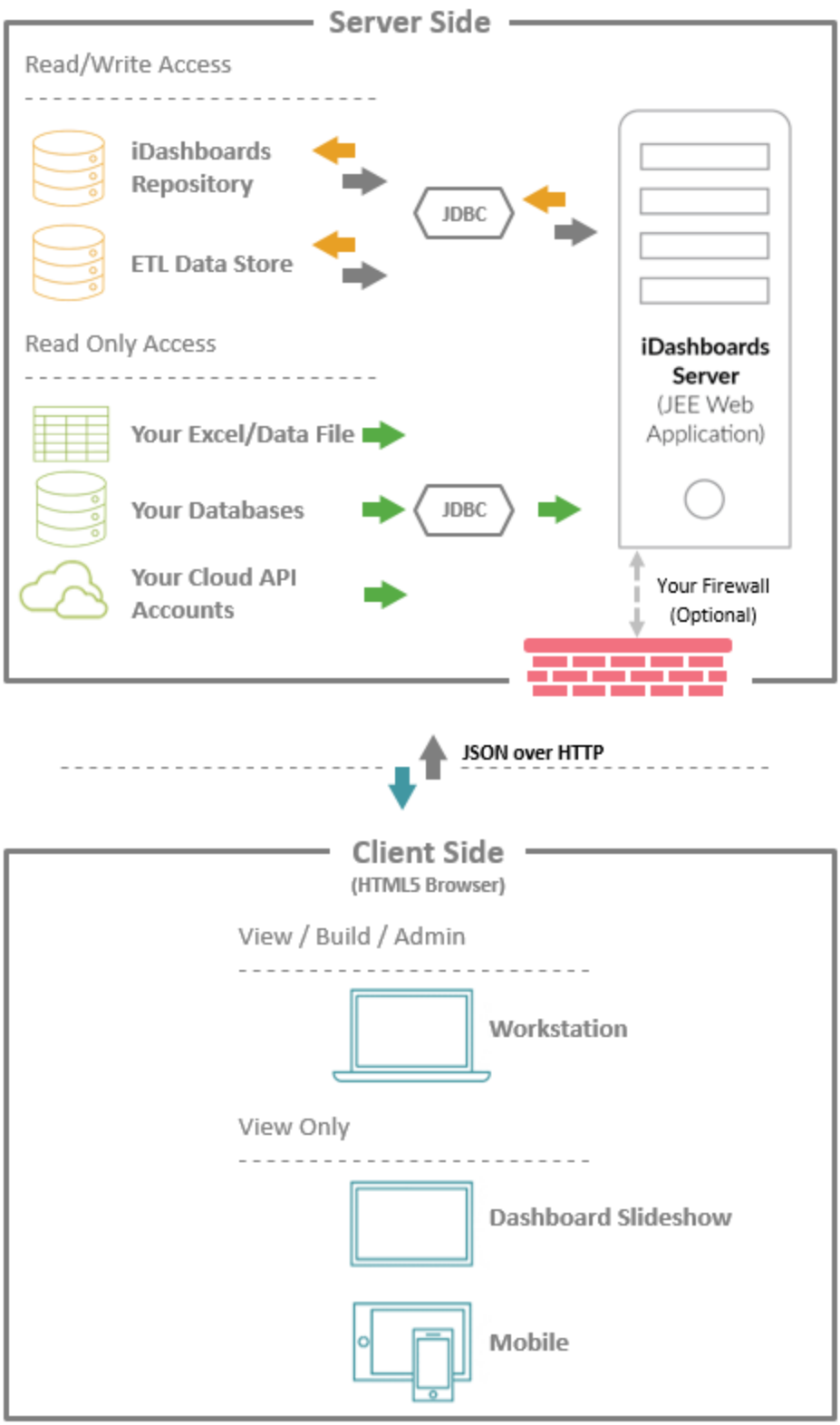
- **Knowledge Base** – Numerous product and technology articles for your review.
- **Community** – Forums and discussion groups for customers to discuss various topics and products amongst themselves.
- **Resources** – Many downloadable resources that can be used with iDashboards.
- **Ideas** – Area for customers to submit feature requests and great product ideas.
- **Blog** – Thoughts, stories and ideas on data and dashboards

3. Application Server Installation

3.1 Architectural Overview

The 4 main components of iDashboards are:

1) The iDashboards Application
    a) HTML5 Browser Interface
        i) View dashboards
        ii) Design charts and dashboards
        iii) Administrator Application
2) The Repository Database
    a) The iDashboards repository database stores application data (metadata) used by iDashboards: such as user, dashboard and chart information. It consists of a small number of database tables that can be created within an existing relational database, or one in a database created specifically for the purpose.
3) The iDashboards Server
    a) The iDashboards Server forms the middle tier between the iDashboards application and the repository database. It is a Java-based, JEE web application, which can be deployed on any JEE-compliant application server such as BEA WebLogic, IBM WebSphere, Oracle Application Server, Apache Tomcat, JBoss and others. The Apache Tomcat server is the supported product and can be downloaded from the internet.
4) The Imported Data Database
    a) Configuring this database will allow the Named Ranges of Excel files, or Delimited and Fixed Column files, to be written into database tables, later to be used as a data source for charts and picklists.
    b) Supported databases include: SQL Server (2005 or later), Oracle (9i or later), MySQL (5.0.3 or later) and PostgreSQL (9 or later).
    c) A database user who can READ, WRITE, CREATE and DROP tables in the database. For example:
        i) **SQL Server**: a user with db_owner privilege OR db_ddladmin AND db_datareader AND db_datawriter privileges
        ii) **Oracle**: a user with Schema owner privilege OR SELECT ANY TABLE, CREATE ANY TABLE, ALTER ANY TABLE, and DROP ANY TABLE privileges
        iii) **MySQL**: ALL object rights, ALL DDL rights
        iv) **PostgreSQL**: Owner of database object OR Superuser role within the database


Server Side
Read/Write Access
iDashboards Repository
ETL Data Store
JDBC
iDashboards Server (JEE Web Application)
Read Only Access
Your Excel/Data File
Your Databases
JDBC
Your Cloud API Accounts
Your Firewall (Optional)
JSON over HTTP
Client Side
(HTML5 Browser)
View / Build / Admin
Workstation
View Only
Dashboard Slideshow
Mobile

3.2 System Requirements

- **Operating System** - Microsoft Windows 2003 or higher or any version of UNIX or Linux for which a Java Virtual Machine is available.
- **Application Server** - Any JEE-compliant application server that supports at least the Servlet 3.0 API and the JSP (Java Server Pages) 2.2 API. Any JEE application server released since 2011 should suffice.
    - **Tomcat (Supported)** – Version 7.0 (or above)

	Tomcat	JDK	Java
Not Supported	6.0	1.5	5
Not Supported	7.0	1.6	6
Not Supported	7.0	1.7	7
Not Supported	8.0	1.7	7
Supported*	8.0	1.8	8
Supported*	9.0	1.8	8
*This version of Java does not support the use of any ODBC connections. While iDashboards can still function with Microsoft Excel files, and Delimited and Fixed Column files; however other data sources requiring ODBC will not function (ex. Microsoft Access).			

    - **Other** – While it is possible to deploy iDashboards using: IBM WebSphere, BEA Weblogic, JBoss or a number of other JEE application servers with the correct Java Virtual Machine (JVM), these application servers cannot be supported by the iDashboards support team.
- **Java** - A Java Development Kit (JDK) depending on the requirements of the application server must be installed on the server. A JDK includes a Java Virtual Machine (JVM) to run the application server, and a Java compiler to compile JSP pages.
    - Version 1.5 (Java 5) – Not Supported
    - Version 1.6 (Java 6) – Not Supported
    - Version 1.7 (Java 7) – Not Supported
    - Version 1.8 (Java 8) – Supported*

**This version of Java does not support the use of any ODBC connections. While iDashboards can still function with Microsoft Excel files, and Delimited and Fixed Column files, other data sources requiring ODBC will not function (ex. Microsoft Access).*

- **Database Server** - Supported repository databases include Microsoft SQL Server (2014 or higher), Oracle (12c or higher), PostgreSQL (9.6 or higher) and MySQL (5.6 or higher). All other repository databases for which JDBC drivers are available can likely be used.

- **JDBC Drivers** - JDBC (Java Database Connectivity) drivers for the target database, and any database used as the Data Source for a dashboard. JDBC drivers are usually bundled with a database, or they can be downloaded for free from the vendor's website.
- **iDashboards Application**
    - **In-Browser using HTML** - Any current browser supporting HTML5 content.
- **iDashboards Auto Uploader**
    - The iDashboards Auto Uploader (IAU) allows for direct uploading of Excel, and Delimited and Fixed Column, files from a client machine to an iDashboards server without requiring the user to log into the iDashboards application. The files can be uploaded immediately using the IAU or they can be scheduled for repeated uploads using a scheduling mechanism.
    - Installation Media and associated documentation:
        - From the iDashboards Administrator application, navigate to Imported Data > Excel Data (or Imported Files) > Auto Uploader
    - Minimum System and User Requirements
        - Microsoft Windows 7 or Windows Server 2008 R2
        - Microsoft .NET 4 Framework – Full profile
        - It also requires that the user running the program to have administrator privileges on the PC for which the IAU is installed. These elevated privileges are necessary to create and execute tasks in the Windows operating system.

3.3 Windows Installation

For new installations of iDashboards, an .EXE executable is available for download on Microsoft Windows operating systems. Do not use the Windows Installer for performing upgrades.

- **Operating System** - Microsoft Windows 2000 or higher
- **Prerequisites**
    - The .EXE executable file from iDashboards.
    - A database server already installed that will be used for the iDashboards repository. This iDashboards production installer supports the following enterprise-class relational database servers: Oracle, Microsoft SQL Server, PostgreSQL and MySQL. Other database types may be utilized upon verification from iDashboards Support.
        - A database created on the database server that will be used for the iDashboards repository. It should be named something logical, such as "idashboards". The SQL scripts for the aforementioned database types are automatically included at the end of the installation.

        - A user account within the database that will be used by iDashboards to connect to the repository. This user account must have SELECT, INSERT, UPDATE and DELETE privileges.
    - A database server for creating the Imported Data Database
        - SQL Server (2005 or later), Oracle (9i or later), MySQL (5.0.3 or later) and PostgreSQL (9 or later).
            - This may be named anything. Best practices for naming the database is ***idb_imported_data.***
            - Prepare a database user with read and write privileges to the Imported Data database.
    - **Google Chrome v70+**: Used in the headless mode, this is only required for the optional feature of generating Dashboard thumbnail images.
        - The Chrome browser does not auto-update while using headless mode. To update Chrome, one must manually launch the browser to begin the update process.
        - Chrome will only be used by iDashboards in headless browser mode

3.4 Manual Installation

The iDashboards server is a JEE web application (http://support.idashboards.com/links/j2eereference), which can be deployed to virtually any JEE compliant application server. A JEE web application is packaged in a WAR (Web Archive) file with a ".war" filename extension. A WAR file is simply a compressed file in the common "zip" format, with a specific internal structure. A WAR file contains most or all of the resources used by a JEE web application, such as HTML files, image files, and Java binary code. The iDashboards WAR file is named "idashboards.war."

This document assumes that a suitable database is already installed and available to be used for the iDashboards repository tables. If a suitable JEE application server is not already installed and available, an Apache Tomcat server is the supported product and can be downloaded from the internet. The remainder of the document assumes an installation of Apache Tomcat web server.

3.4.1 Installation Roadmap

The exact procedure for installing iDashboards will vary from one installation to the next, since a wide variety of databases or app servers may be used. Once the database and the app server have been installed, the basic steps for installing iDashboards are:

1. ***Create the iDashboards repository tables*** – This step should be performed by your organization's DBA (database administrator). See Section 3.4.2, "Creating the iDashboards Repository Tables", for more information.

2. **Create the ivizgroup home directory** – This is a directory where iDashboards stores needed information. See Section 3.4.4, "Creating the ivizgroup home directory", for more information.

3. **Configure the ivizgroup.properties file** – This is a text file read by iDashboards, which contains information needed to connect to the repository database. See Section 3.4.5, "Configuring ivizgroup.properties", for further information.

4. ***Install the iDashboards license file (optional)*** – If using a License File, instead of a License Key, the iDashboards license file, named "idashboards.lic", is provided. It can be installed at this point. See Section 3.4.6, "Installing the License File", for more details. Otherwise Section 4.3 "Activating iDashboards License" covers licensing when the Administrator first logs into the application.

5. ***Deploy the idashboards.war file to the application server*** – See Section 3.4.7, "Deploying the Application" for more information.

3.4.2 Creating the iDashboards Repository Tables

DDL (data definition language) scripts for creating the iDashboards repository tables are located in the **scripts/database** directory of the installation CD. The database administrator should be familiar with how to execute scripts on the target database; if not, the database documentation should be consulted.

Repository creation scripts have been provided for some of the more popular databases in both standard (non-Unicode) and Unicode database format. These scripts are to be used when creating the iDashboards repository for the first time. Standard scripts are located in the scripts/database/standard/create_repository folder. Databases that support Unicode have scripts located in the scripts/database/unicode/create_repository folder. The folder and script names clearly indicate the type of database they are intended for. The script "generic_standard_create_repository.sql" is written using ANSI SQL data types and syntax, and should work with most other databases, possibly with minor modifications. Unicode repositories are only supported on SQL Server, Oracle and PostgreSQL.

Repository upgrade scripts have been provided for some of the more popular databases in both standard (non-Unicode) and Unicode database format. These scripts are to be used when upgrading iDashboards from a prior version to the current version. Databases that support Unicode have scripts located in the scripts/database/standard/upgrade_repository subfolders. Unicode scripts are located in the scripts/database/unicode/upgrade_repository subfolders. The folder and script names clearly indicate the type of database they are intended for. The scripts "generic_standard_upgrade_XX_to_YY.sql" (where "XX" is the old version and "YY" is the newer/current version) are written using ANSI SQL data types and syntax and should work with most other databases, possibly with minor modifications. Only run the single upgrade script that will update your current installed version to the latest release version. For example, if you have iDashboards v9.7 installed, only run the upgrade script "<database_type>_<standard_or_unicode>_upgrade_97_to_YY.sql", where YY is the newer/current version of iDashboards.

Repository Guidelines – The following points should be noted when creating the repository tables:

1. It is recommended that the repository tables are created in their own separate schema, and a separate user account is created with which iDashboards can connect to the database.
2. When iDashboards queries or updates repository tables, table names are not qualified with their schema name in SQL statements. Therefore, the schema selected to hold the repository tables and the database user account used to connect iDashboards to the database should reflect this fact.
3. The supplied DDL scripts do not grant any privileges on the created tables; however the user account with which iDashboards accesses them must have SELECT, INSERT, UPDATE and DELETE privileges on those tables.

3.4.3 Creating the Imported Data Database

In addition to the iDashboards Repository database, the Imported Data functionality requires an additional database.

- A database server for creating the Imported Data database
    - SQL Server (2005 or later), Oracle (9i or later), MySQL (5.0.3 or later) and PostgreSQL (9 or later).
        - This may be named anything. Best practices for naming the database is ***idb_imported_data***
        - Prepare a database user with read and write privileges to the Imported Data database.

To configure the Imported Data database, see Section 11.2, "Configure the Imported Data Database".

3.4.4 Creating the ivizgroup home directory

The ivizgroup home directory (also referred to as a "folder") is where various files needed by the iDashboards server application, such as the license and configuration files, are kept. In order for iDashboards to function properly, this directory must exist and be readable.

The ivizgroup home directory must be manually created as part of the iDashboards installation. The default location is in the home directory of the user account under which the iDashboards application server is running, and the default name is "ivizgroup" (all lowercase). So for example, on a Windows system where the iDashboards application server is running as the Windows user "tomcat", the ivizgroup home directory would be **C:\Documents and Settings\tomcat\ivizgroup**.

The default location of the ivizgroup home directory can be overridden by setting a Java system property called "ivizgroup.home" to the path of the ivizgroup home directory.

A Java system property is set on the command line that is used to start a Java Virtual Machine. For many JEE application servers, this command line will be contained within a startup script.

Note: A system property is set with the switch " Dname=value", so for example, an alternate ivizgroup home directory might be set with the switch " Divizgroup.home=C:\ivizgroup".

Note: Throughout the remainder of this document, <IVIZGROUP HOME> will be used to indicate the ivizgroup home directory.

After the location of the <IVIZGROUP HOME> directory has been established and created, three subdirectories must be created within it:

1. <IVIZGROUP HOME>\config — This directory is where iDashboards configuration files are kept.
2. <IVIZGROUP HOME>\drivers — JAR files containing JDBC drivers (explained elsewhere in this document) can be stored in this directory, making them available to iDashboards at runtime.
3. <IVIZGROUP HOME>\logs — This directory is where iDashboards log files are written.

3.4.5 Configuring ivizgroup.properties

The ivizgroup.properties file is a plain text file read by the iDashboards Server application. It contains the information iDashboards needs to connect to the repository database, and the settings that control runtime logging (see Section 13.4.3 Log Configuration).

Runtime location of ivizgroup.properties – By default, iDashboards will look for the ivizgroup.properties file in the <IVIZGROUP HOME>\config directory. Both the name and the location of ivizgroup.properties can be overridden, however, by setting a Java system property called "ivizgroup.properties" to the path and filename of the alternate file.

The ivizgroup.properties file is in "Java properties" format, which has the following characteristics:

1. Blank lines, and lines whose first non-whitespace character is # or ! are ignored. (Lines beginning with # or ! can be used for comments)
2. Other lines should adhere to the format:

   ```
   name=value
   ```

   where "name" is the name (unique within the file) of a particular property, and "value"

is the value assigned to that property. Property names and values are both case-sensitive.

3. Although the Java properties format allows for leading and trailing whitespace in property names and values (when properly escaped), neither property names nor values in ivizgroup.properties should contain leading or trailing whitespace.

A template ivizgroup.properties file is located in the **config** directory of the installation CD. The names of the properties needed by iDashboards are included in the supplied file, along with instructional comments.

Configuration overview – iDashboards actually uses a pool of at least three connections to the repository database, so multiple users can access it simultaneously. There are two possible ways iDashboards gets access to a pool of repository database connections:

1. ***Server-Managed Connection Pool*** — iDashboards uses a connection pool created and managed by the JEE application server. (This may be referred to as a "DataSource" in the server's documentation.) This option is for advanced users.

2. ***iDashboards-Managed Connection Pool*** — iDashboards creates and manages its own connection pool. This is the recommended option for most users.

Using a server-managed connection pool – If a server-managed connection pool is used, it must be accessible through the server's naming and directory service, which can also be referred to as the server's JNDI (Java Naming and Directory Interface) service. Such a connection pool is given a "JNDI Name", which web applications can use to access it. To use a server-managed connection pool, only a single setting is needed in ivizgroup.properties:

```
db.jndiName=<JNDI name of server-managed connection pool>
```

An example entry might look like:

```
db.jndiName=idbRepository
```

3.4.5.1 Using an iDashboards-managed connection pool

It is important to note that the presence or absence of the **db.jndiName** property in ivizgroup.properties determines whether or not iDashboards will use a server-managed connection pool. If such an entry is present, iDashboards will always try to "look up" and use a connection pool by that name, and it will fail if one is not available. **If iDashboards is to create and use its own connection pool, then the db.jndiName setting must be removed or commented out of ivizgroup.properties.**

To make iDashboards create and manage its own pool of connections to the repository database, four settings are needed in ivizgroup.properties:

```
db.user=<database username to connect with>

db.password=<database password to connect with>
```

```
db.driverClass=<The Java class name of the JDBC driver to use>

db.url=<The URL of the repository database>
```

Additionally, three optional properties may be specified:

```
db.maxConnections=<maximum size the connection pool may grow to>

db.validateConnections=<true or false>

db.password.encrypted=<true or false>
```

The **db.user** and **db.password** properties are the credentials used to connect to the repository database. As mentioned previously, the database user account that iDashboards connects with must have SELECT, INSERT, UPDATE and DELETE privileges on the repository tables.

The **db.driverClass** and the **db.url** properties depend on the type of database and the JDBC drivers used to connect to it. The documentation for the JDBC drivers should be consulted when entering values for these properties; however, this document contains sample entries for some of the more popular databases.

The **db.maxConnections** property indicates the maximum number of connections that will be created by an iDashboards-managed connection pool. If this property is missing or blank, a default value of 20 will be used. Connections will only be created and added to the pool on an as-needed basis, so the maximum number of connections may never be reached unless the iDashboards server is extremely busy.

The **db.validateConnections** property indicates whether or not connections should be tested before they are used. The default value is true. If true, then a small test query will be run on each connection when it is taken from the connection pool for use, and if it fails, the connection will be considered "dead" and removed from the connection pool. This can prevent users from seeing error messages that will occur when the repository database goes temporarily offline, leaving disconnected, or dead, connections in the pool. Since there is a small performance cost associated with testing connections, this property can be set to false in cases where the repository database is unlikely to go offline while the iDashboards server remains online. If it is set to false, however, dead connections will remain in the pool and cause errors until the iDashboards server is restarted.

The **db.password.encrypted** property is set to true to indicate that the password set with the **db.password** property has been encrypted with the idb_encrypt tool. This is a command line tool shipped with iDashboards that can encrypt a password, so it can be included in the ivizgroup.properties file without being revealed to unauthorized persons who may have access to the file. If the value for **db.password.encrypted** is anything other than "true" (case-insensitive), then the password is assumed to be in cleartext. For information on encrypting passwords with idb_encrypt (see Section 6.1, "Using the idb_encrypt tool").

Troubleshooting – The first step in troubleshooting problems with ivizgroup.properties is to make sure that iDashboards is able to locate it at runtime. You won't be able to login and

use iDashboards until ivizgroup.properties required database connection properties have been properly configured. Upon successful log-in, you will be presented with the iDashboards Lobby. From there you can access the Administrator Application by selecting the "ADMIN" icon. The under menu item "System > About", the path to the ivizgroup.properties file is displayed in the "Deployment Information" section, along with a "Reload" link that will reread the contents of the file.

Note: The "Reload" link should only be used while troubleshooting problems with the ivizgroup.properties file or connecting to the repository database. Once a connection to the repository database has been established, hitting the Reload link will have no effect on the current connection, and any changes to the connection properties will require a server restart to take effect.

3.4.6 Installing a License File

If using a license file, instead of a "License Key", the iDashboards license file, named "idashboards.lic", should be copied to the <IVIZGROUP HOME> directory. The default location of the license file, however, can be overridden. iDashboards looks for the license file in the following locations, in the order listed, and the first one it finds will be the one used:

1. It will search the application server's Java classpath for a file named idashboards.lic.
2. It will check for a Java system property called "idashboards.license", which, if present, must be set to the full path (including filename) of the idashboards.lic file.
3. It will check in the <IVIZGROUP HOME> directory for a file called "idashboards.lic."
4. It will check in the current working directory of the JEE application server for a file named idashboards.lic.

Note: An explanation of the Java classpath is beyond the scope of this document, and installing the idashboards.lic file in the Java classpath is not recommended.

After iDashboards has located and read the license file, its location will be displayed near the bottom of the iDashboards administrative login screen.

Note: The standard, preferred way to deploy the iDashboards license file is in the <IVIZGROUP HOME> directory, so the server can locate it as described in 3 above. Other locations should be used only under special circumstances, under the direction of iDashboards Support.

3.4.7 Deploying the Application

The procedure used to deploy the idashboards.war file depends on the type of JEE application server used. Since iDashboards can be installed on a variety of different application servers, it would be impossible to document here the exact steps required for each one. Deployment of a WAR file is usually a straightforward process, however, which

should be thoroughly explained in the application server's documentation. The process may involve copying the WAR file to a specific directory and restarting the server, or uploading the WAR file to the application server through a web interface. Some manual editing of server configuration files may be required.

The idashboards.war file is located in the **bin** directory on the installation CD.

Choosing a Context Root – Regardless of the application server used to host iDashboards, the iDashboards web application must be assigned a "context root" within the server's URL space. Conceptually, the context root can be thought of as a subdirectory beneath the server's root URL, which forms the root of the web application's URL space. This allows multiple web applications from different sources to be deployed to the same application server without URL conflicts.

It is recommended that "/idashboards" be used as the context root for the iDashboards web application. This means that if iDashboards is deployed on "intranet.mycompany.com", the URL for the iDashboards application would be *http://intranet.mycompany.com/**idashboards***, and the iDashboards Administrator application would be accessed from the iDashboards Lobby by selecting the "ADMIN" tile.

Moreover, since the iDashboards WAR file is named idashboards.war, some application servers (such as Tomcat) will automatically default its context root to "/idashboards", so choosing that can simplify the deployment process.

3.4.8 JDBC Driver Configurations

This section provides information to help determine the **db.driverClass** and **db.url** property settings in the ivizgroup.properties file. As mentioned in Section 3.4.5, "Configuring ivizgroup.properties", the values used for these properties depend on the target database and the JDBC drivers used to connect to that database.

If documentation is available for the JDBC drivers to be used, it should be consulted. However, guidelines for some of the more popular databases and their JDBC drivers are provided at the end of this section.

JDBC drivers are usually packaged in a JAR (Java ARchive) file, which has a ".jar" filename extension. A JAR file is simply a compressed file in the common "zip" format. Some vendors package their JDBC drivers in a regular zip file; for instance, Oracle JDBC drivers are commonly distributed in a file named "classes12.zip." For the purposes of this document, however, these files will be generically referred to as JAR files.

Obtaining JDBC drivers – JDBC drivers are available for virtually all enterprise-class relational databases, and in most cases they can be downloaded for free from the database vendor's website. (They may also come bundled with the database itself.) If you need to obtain JDBC drivers for your database, a good place to start (aside from the vendor's website) is Sun's database of JDBC drivers, which can be searched or browsed at:

http://support.idashboards.com/links/sunjdbcdrivers

Installing JDBC Drivers – In order to be used by iDashboards, a JDBC driver's JAR file should be placed in the <IVIZGROUP HOME>\drivers directory. Alternatively, they may be added to the classpath of the application server. If the application server is already in production and hosting applications that connect to the target database, this step has probably already been done. The procedure for modifying an application server's classpath will vary from one application server to another and is beyond the scope of this document, so the application server's documentation should be consulted.

The Driver Class – Each JDBC driver contains a unique "driver class", which can be thought of as the front door through which applications access the driver's functionality. To use a JDBC driver, an application must be told the name of its driver class, and that is the purpose of the **db.driverClass** property in ivizgroup.properties. A driver class name may look odd to someone who is not a Java programmer; a typical example is Oracle's, which is "oracle.jdbc.driver.OracleDriver". Consult the documentation for your JDBC driver to determine the name of its driver class, or check the "Example Settings" section below to see if it's listed there.

When it starts up, iDashboards searches the <IVIZGROUP HOME>\drivers directory and the application server's classpath for each one of a list of common JDBC driver classes, and each one it finds is available by selecting the "Installed JDBC Driver" link from "Data Sources" screen in the Administrator Application. If your driver class name appears in this list, it means the driver's JAR file has been successfully installed. It's important to note, however, that if your driver class does not appear in the list, it does not necessarily mean that it is not properly installed; it may be because it is not on the list of drivers that iDashboards searches for. This list is available in the file config/jdbc_drivers.txt on the iDashboards installation CD.

The Database URL – Another key piece of information iDashboards needs to connect to its repository database is the URL of the database, and that is provided to it by the **db.url** property in ivizgroup.properties. The URL (Uniform Resource Locator) is a character string containing all the information the JDBC driver needs to locate and connect to the database.

A database URL always begins with "jdbc:", but beyond that, the syntax differs from one JDBC driver to the next. URLs generally consist of segments separated by colons (:), slashes (/) or other characters. The information they contain usually includes the hostname of the database server and the name of the database itself, possibly the port number on which the server is listening, and possibly (but not usually) the username and password used to make the connection.

Consult the documentation for your JDBC drivers to determine how to construct the URL that must be set as the **db.url** property in ivizgroup.properties, or see if a sample URL is provided in the "Example Settings" below.

3.4.8.1 Example Settings

This section lists the driver class names and sample URLs for some of the more popular databases. In the sample URLs, the following words should be substituted with the given values:

hostname – This is the network name of the server on which the target database is running. Some JDBC drivers will also accept an IP address in place of the hostname.

database – This is the name of the target database. (In the case of Oracle, it would be the SID)

port, or ***nnnn*** (where *nnnn* is a string of digits) – This is the port number on which the database server listens for connections. In the samples below, if the default port number for the indicated database type is known, it is used in the sample URL. (For example, the default port for Oracle is 1521.) If it is not, then the word "port" is used as a placeholder, which should be replaced with the correct port number. In either case, the correct port number should be substituted where appropriate. Note that for many JDBC drivers, the port number can be omitted from the URL if the database is listening on its default port.

dbusername – This is the name of the database user that iDashboards will connect as. In most cases, this will not be part of the URL.

dbpassword – This is the password that iDashboards will use to connect to the target database. In most cases, this will not be part of the URL.

Following are example settings for some popular databases. This list is by no means complete, nor is it necessarily accurate for all versions of the listed databases. It is only intended as a starting point, and the documentation for a particular JDBC driver should be considered the authoritative source.

3.4.8.2 Microsoft SQL Server (Microsoft driver)

db.driverClass: com.microsoft. sqlserver. jdbc.SQLServerDriver

db.url:
jdbc:sqlserver://hostname:1433;databaseName=database;selectMethod=cursor

Note: If a named instance is being used, append

;instanceName=name

to the URL (substituting the instance name for "name") and use the port number of the SQL Server Browser service (which by default is 1434), or use the port number on which the named instance listens for incoming connections.

3.4.8.3 JTDS Driver for Microsoft SQL Server

http://support.idashboards.com/links/sourceforgejtds

db.driverClass=net.sourceforge.jtds.jdbc.Driver

db.url= jdbc:jtds:sqlserver://hostname:1433/database;TDS=8.0;

Note: Use "TDS=4.2" for SQL Server 6.5 and "TDS=7.0" for SQL Server 7.0.

3.4.8.4 Oracle (OCI driver)

db.driverClass=oracle.jdbc.driver.OracleDriver

db.url= jdbc:oracle:oci8:@database

3.4.8.5 Oracle (Thin driver [preferred])

db.driverClass=oracle.jdbc.driver.OracleDriver

db.url= jdbc:oracle:thin:@hostname:1521:database

3.4.8.6 PostgreSQL

db.driverClass=org.postgresql.Driver

db.url= jdbc:postgresql://hostname:port/database

3.4.8.7 Sybase Adaptive Server Enterprise

db.driverClass=com.sybase.jdbc2.jdbc.SybDriver

db.url= jdbc:sybase:Tds:hostname:port/database

3.4.8.8 Sybase jConnect

db.driverClass=com.sybase.jdbc3.jdbc.SybDriver

db.url=jdbc:sybase:Tds:hostname:port?ServiceName=database

3.4.8.9 MySQL Connector/J

db.driverClass=com.mysql.jdbc.Driver

db.url=jdbc:mysql://hostname:port/database

3.4.8.10 Caché

db.driverClass= com.intersys.jdbc.CacheDriver

db.url= jdbc_jdbc.url=Cache://machine:1972/namespace

3.4.9 Deploying iDashboards to Tomcat

iDashboards will support a Tomcat application server (see Section 3.2, "System Requirements" for Tomcat version requirements).

This section describes how to deploy the iDashboards WAR file (idashboards.war) to a Tomcat application server. In this section, <TOMCAT HOME> refers to the directory where Tomcat has been installed and <IVIZGROUP HOME> refers to the location of the ivizgroup home directory. The Apache Tomcat server is a supported product and can be downloaded from the internet.

It is also assumed that the repository database tables have been created as described in Section 3.4.2, "Creating the iDashboards Repository Tables", and the proper JDBC drivers have been acquired.

All of the following instructions should be performed on the computer on which Tomcat is installed:

1. If the Tomcat server is running, stop the service.
2. Copy the file **bin\idashboards.war** from the iDashboards installation CD to <TOMCAT HOME>\webapps.
3. Copy the file **config/ivizgroup.properties** from the installation CD to <IVIZGROUP HOME>\config.
4. Configure <IVIZGROUP HOME>\config\ivizgroup.properties as described in Section 3.4.5, "Configuring ivizgroup.properties".
5. Copy the license file, idashboards.lic (provided separately from the installation CD) to <IVIZGROUP HOME>.
6. Copy the JAR files for any required JDBC drivers to <IVIZGROUP HOME>\drivers.
7. Restart the Tomcat server.
8. To verify that iDashboards is running correctly, start a browser and open the login screen, which should be available at:

   `http://localhost:8080/idashboards`

9. If the login screen appears, attempt to log into with the username "admin" and the password "change_me". If iDashboards has successfully connected to the repository database, the login should succeed taking you to the iDashboards Lobby.
10. From the Lobby select "ADMIN" for the Administrator application, and then menu item "System > About". Verify the values under "License Information" and "Deployment Information", including the path to the ivizgroup.properties and the idashboards.lic files.

4. Starting the iDashboards Administrator Application

The iDashboards Administrator application is a browser-based application used to configure an iDashboards Server. It is available once the idashboards.war file has been deployed to the application server and properly configured. Some key tasks that may be performed through the iDashboards Administrator application are:

- Creating chart/dashboard Categories
- Creating Users, User Groups and assigning privileges
- Creating links to external data sources
- Managing system settings

4.1 Determining the URL of the Application

The URL of the iDashboards Server is the web address used to access the iDashboards application with a web browser. To determine it, you must know three things:

1. The hostname or IP address of the server on which iDashboards is deployed.
2. The port number on which the iDashboards application server is listening, if it is other than port 80, which is the default HTTP port number.
3. The "context root" under which the iDashboards application was deployed. Normally this will be "idashboards." See Section 3.4.7, "Deploying the Application" for more information.

Once these three components are known, the URL of the iDashboards application will be:

```
http://<servername or IP address>:<port number>/<context root>
```

This will open the iDashboards login screen.

For example, if the hostname is "dashmachine", the port number is 8080, and the context root is "idashboards", the URLs for iDashboards would be:

Interface	URL
► **iDashboards Application**	`http://dashmachine:8080/idashboards`
Data Hub Application	`http://dashmachine:8080/idbdata`
If the port number is 80, it can be omitted from any of the URLs, as seen below:	
► **iDashboards Application**	`http://dashmachine/idashboards`

► = Topic included within this manual

In some cases, such as when the server is being accessed over the Internet, it may be necessary to append a domain name to the hostname:

```
http://dashmachine.mycompany.com/idashboards
```

When the URL is accessed through a web browser, the login screen should appear.

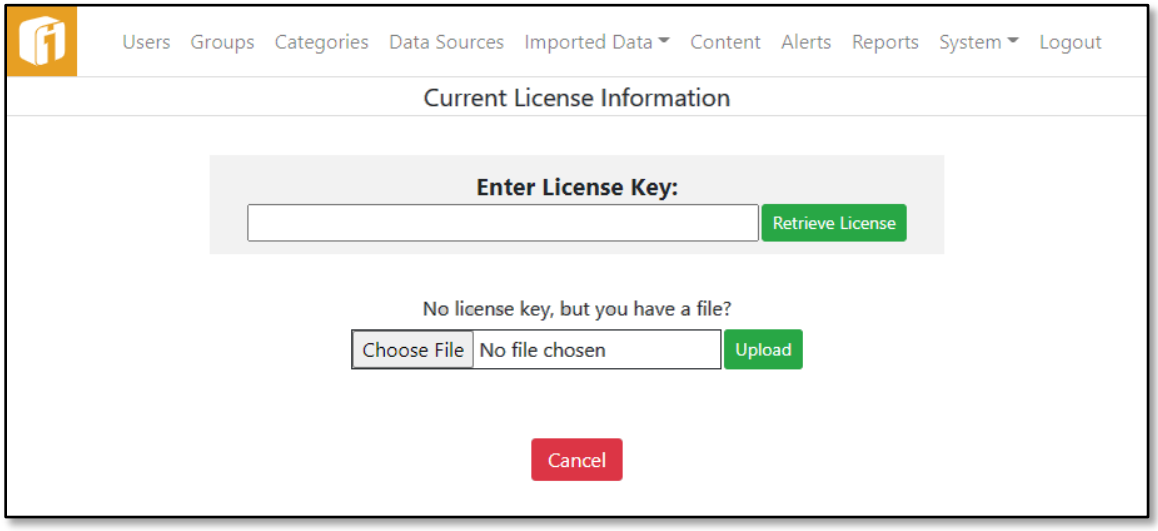
4.2 Logging into the iDashboards Application

When iDashboards is initially installed, an administrative user account is created with the username *admin* and the password *change_me*. These credentials can be used to log into the iDashboards application through the login screen.

Note: It is highly recommended that the admin password is changed after installation.

4.3 Activating iDashboards License

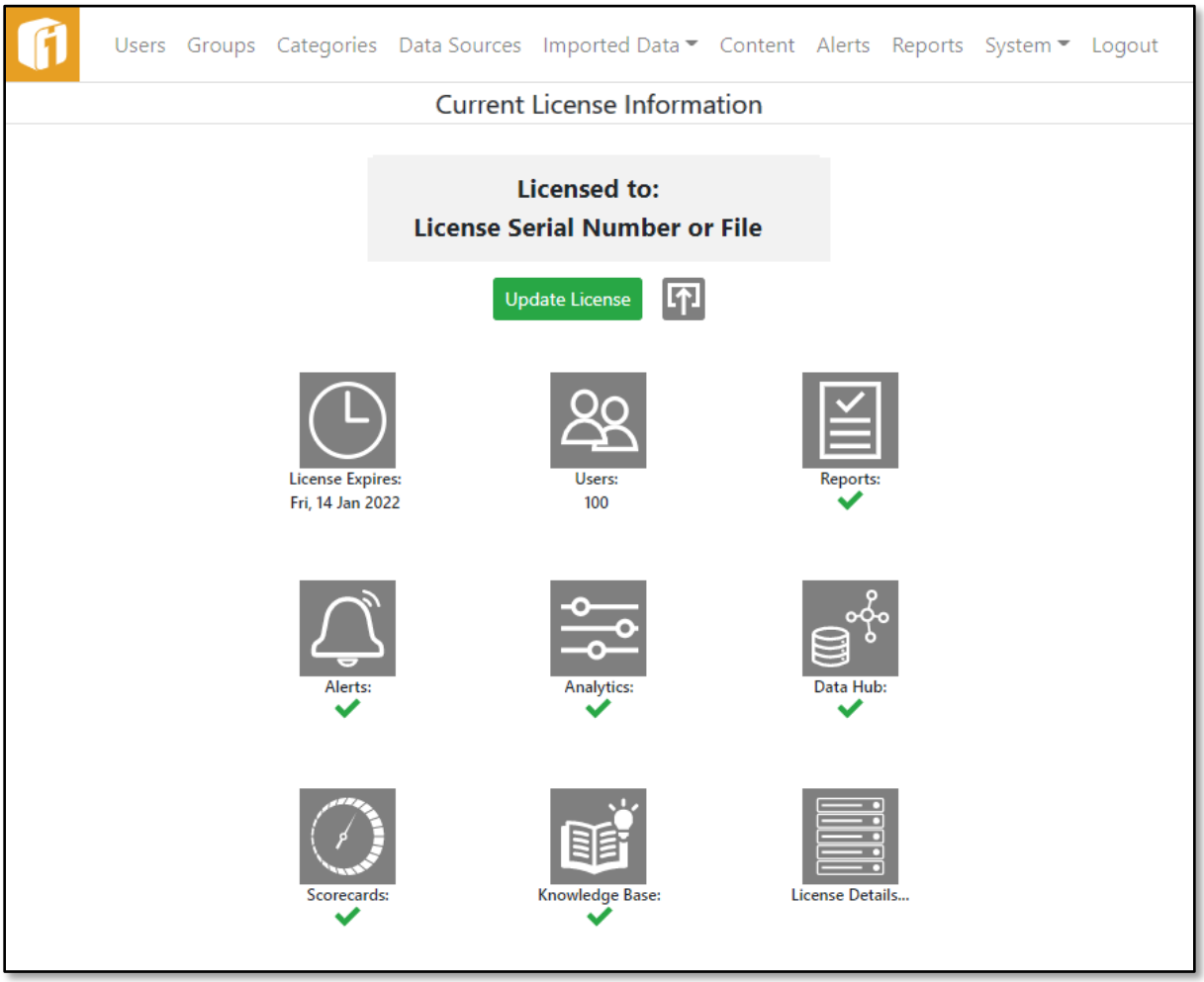
If a license file has not been setup as part of a manual installation, as covered in Section 3.4.6 "Installing the License File", the initial login will launch the licensing page. Here there are two options for activating the iDashboards license.



1. License Key
    a. Enter assigned License Key and select "Retrieve License" button.

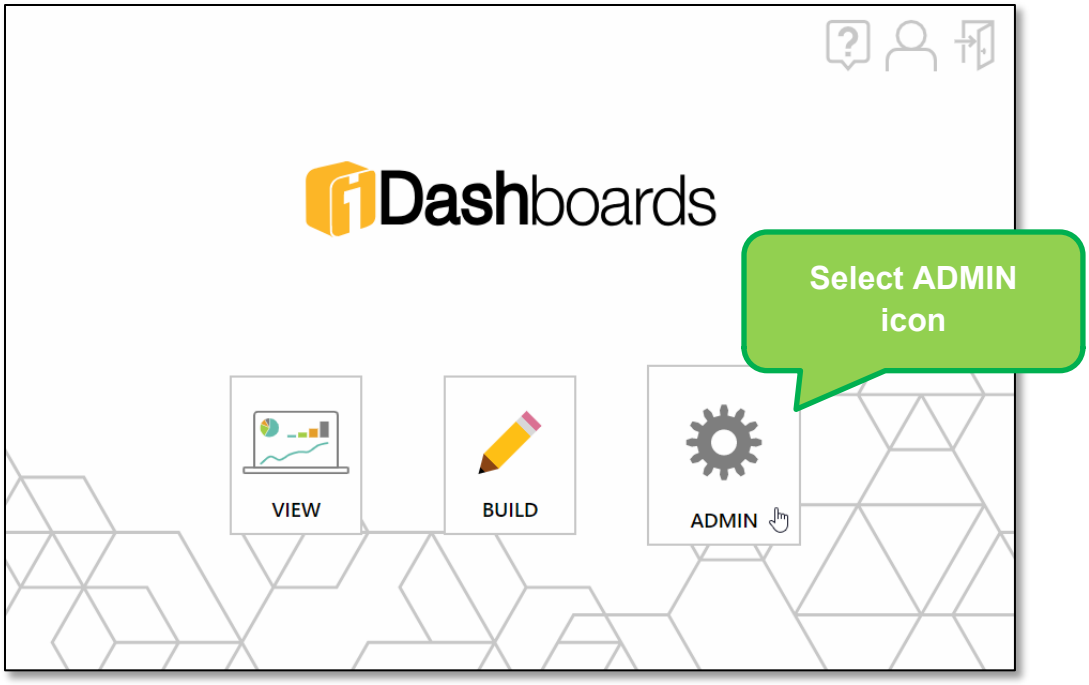
2. License File
    a. Use the "Choose File" button to locate and select the provided License File (i.e. idashboards.lic).
    b. Once selected, use the "Upload" button.

Either way, once activated the Current License Information page will show the status of modules as defined by the license, and the option to view and copy license details.



4.4 The Administrator Application Home Screen

Upon successful login, you are in the iDashboards Lobby. From there you can access the Administrator application by selecting the "ADMIN" tile.



The Administrator application Home screen will be displayed. The Administrator application Home screen contains a menu bar, as well as tiles, for accessing the other administrative screens.

4.5 Leaving the iDashboards Administrator Application

To leave the iDashboards Administrator application, select "BACK" tile. You will be returned to the iDashboards Lobby.

4.6 Logging out of iDashboards Application

From the iDashboards Lobby, to log out of the iDashboards application, select the "Logout" () icon, from the top right corner.

From any iDashboards Administration screen, to log out of the iDashboards application, select "Logout" on the menu bar.

5. Temporary Lockdown of iDashboards

All of iDashboards can be placed into a "locked down" state, during which only the iDashboards system user (the "admin" account) can access the application. Other users that are already logged into the application when a server is locked down will be logged out if they attempt any activity that requires server access. Switching the server into or out of the locked down state does not require a server restart.

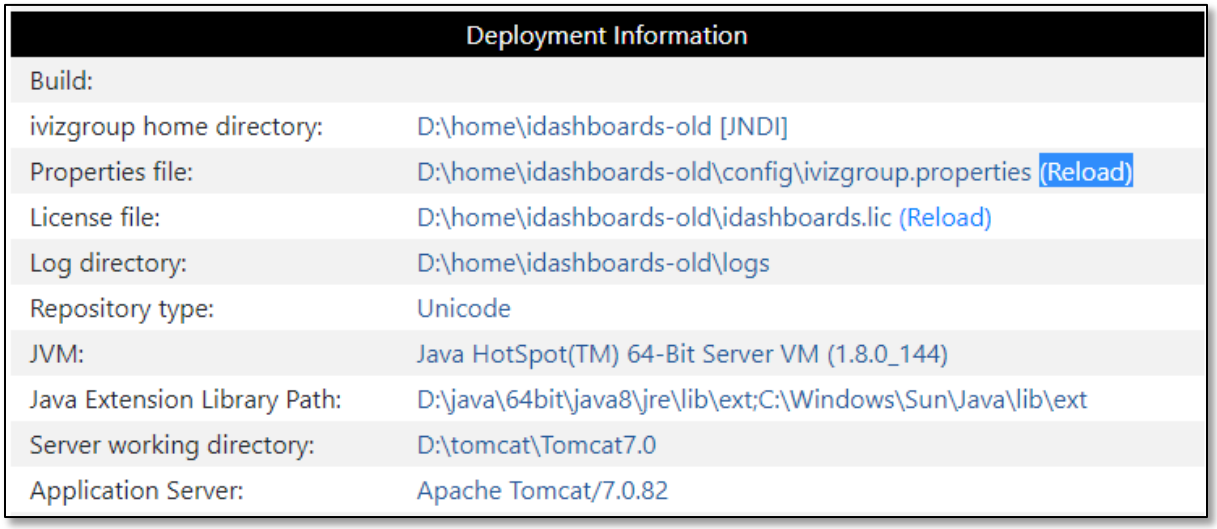
The following steps will lock down an iDashboards server:

1. In the ivizgroup.properties file, add or update the following property:

   ```
   server.lockdown=true
   ```

2. Save the ivizgroup.properties file.

On the "About iDashboards" screen of the Administrator application, under "Deployment Information", select the "Reload" link next to the location of the ivizgroup.properties file, as shown here:



The same steps are used for unlocking a locked down server, except the server.lockdown property should be commented, removed, or changed to a value other than "true".

6. Security Overview

In iDashboards, information is displayed to an end user through a dashboard. A dashboard consists of one or more charts, and each chart displays a specific type of information, such as sales or production figures. It is often desirable to limit the amount or type of information a particular user may view or change, and this is accomplished through iDashboards' security framework. The major components of iDashboards' security framework are *users, roles, groups and categories*.

A **user**, as the name implies, is someone who has been granted access to the iDashboards application. Each user has a login ID (also referred to as a username), a password, and a user record in the repository database.

Each user is assigned one of the following **roles** in the iDashboards system:

1. Admin
2. Data Admin
3. Builder
4. Viewer

A user's role does not determine *what* charts or dashboards a user has access to as much as it determines what that user *may do* with those charts or dashboards, for example, creating dashboards and charts through the iDashboards Build interface, or only viewing dashboards and charts through the iDashboards View interface. The capabilities available to each role are explained in detail in Chapter 7 "Managing Users".

A **group** is what group's logically-related users together. Each iDashboards user must belong to one group and a user can belong to more than one group.

A **category** groups related charts and dashboards together. Each chart and dashboard must belong to *exactly one* Category. A chart, however, may be a component of multiple dashboards, even dashboards in a Category other than its own. The importance of this point in determining what a user may view or change will be explained below.

A group may be granted *view* or *save* access to multiple categories. View access means that users in the group may see dashboards and charts from that Category. Save access means those users may also modify and save changes to those dashboards and charts.

When a user logs into the iDashboards View and Build interfaces, a list is built of all categories to which that user's groups have access, along with the dashboards in each category. These categories are sometimes referred to as the user's *accessible categories*. The user may view all of the dashboards in those categories, and modify those in categories to which at least one of the user's groups has save access. If a dashboard contains a chart that is not in the user's accessible categories, it will still be displayed on the dashboard, because the dashboard's access level takes precedence over the charts.

6.1 Using the idb_encrypt tool

The idb_encrypt tool is a Java-based command-line utility than can encrypt passwords in a format that iDashboards can decrypt. It is used to encrypt passwords that are stored in the ivizgroup.properties file, specifically the database password that is set with the **db.password** property. Passwords that are part of a database URL should not be encrypted with idb_encrypt.

It is not necessary to encrypt the database password stored with **db.password** in ivizgroup.properties, but it may be desirable if the password should not be revealed to persons who must access that file for other reasons. It is important, however, that if the database password in ivizgroup.properties is encrypted, then the **db.password.encrypted** property must be set to "true" (without the quotation marks.) If the database password is not encrypted, then the **db.password.encrypted** property should be deleted, commented out, or set to "false".

idb_encrypt is contained in the file **tools/idb_encrypt.jar** on the iDashboards installation CD. It can be used from either a UNIX or Windows command line, on any machine that has a Java Virtual Machine (JVM) installed. To use it, copy idb_encrypt.jar from the CD to the current directory of your command shell, and execute the following command, substituting the password to be encrypted for <password>:

```
java –jar idb_encrypt.jar <password>
```

The encrypted password will be printed to the console, from which it can be copied into ivizgroup.properties. For example, this command:

```
java –jar idb_encrypt.jar mysecret
```

will produce this output:

```
6c7b776d735225f4
```

which can be copied into ivizgroup.properties like this:

```
db.password=6c7b776d735225f4

db.password.encrypted=true
```

If the java command is not recognized by the command shell, you may have to enter the full path to the java executable file, and enclose it in quotations marks if it contains spaces, as this example shows:

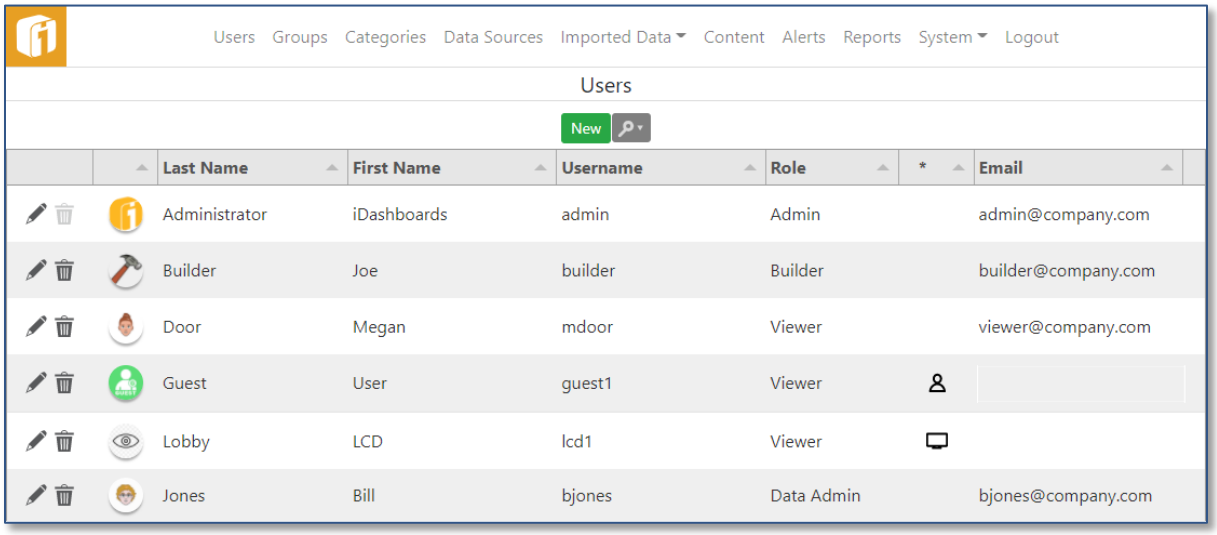
```
"C:\Program Files\Java\jdk1.5.0_03\bin\java" –jar idb_encrypt.jar
mysecret
```

7. Managing Users

Each user requiring access to iDashboards must have a user account. User accounts are managed in the "Users" screen of the Administrator application. To access the User screen, select the "USERS" tile from the Administrator home screen, or "Users" on the menu bar.

The "Users" screen displays the list of up to 1000 iDashboards users.

> *Note: The default sort order of the list is by "Last Name", however, a different sort column can be used by selecting the column title and toggling ▾ascending or ▴descending.*



7.1 The iDashboards System User

When iDashboards is installed, a user account with the username "admin", the password "change_me" and the Admin role is created. This account, which cannot be deleted, is referred to as the "iDashboards system user." All of the properties for this user account, except the username and role, can be changed.

> *Note:* It is highly recommended that the "admin" password is changed after installation.
>
> *Note: Because the properties for this user account can be changed, and the fact that this user account cannot be deleted, make note that it is possible to have an account appear as unremovable due to this condition.*

7.2 Adding a User

To create a new user, select the "New" button and then enter the user's first name, last name, user name (login ID) and password into the appropriate textboxes. Select a role for the user from the "Role" dropdown.

- **Viewer** – Users assigned to this role have read-only access to dashboards within the categories where they have permissions. They can, however, fully interact with charts and dashboards. They do not have a "Personal" category and there are limited menu options.
- **Builder** – Users assigned to this role have full dashboard, chart and picklist building capabilities within the categories where they have "save" permissions. They can upload files to the "Content" directory and they do have a "Personal" category.
- **Data Admin** – This role allows all Builder role functions, plus the ability to log into the Data Hub for performing data manipulations, along with Excel, and Delimited and Fixed Column, file uploads.
- **Admin** – Users assigned this role have all the permissions of the Data Admin role and are the only users that can log into the Administrator Application and perform administrative functions.

Once the basic user information has been entered, select the "Save" button to add the new user. The newly-added user will appear in the list, with location based on the sort-order in effect.

Note: It is also possible to add a profile image to the user, by selecting the default user image and selecting an image to replace it.

Values entered for a new user must adhere to the following rules:

- **Last name** — 1 to 50 characters.
- **First name** — 1 to 50 characters.
- **Username** — 1 to 75 characters. The username must be unique within the system, without regard to case. In other words, "ADMIN" would be considered the same as "admin." However, when logging into the system, a user should assume the username is case-sensitive.
- **Password** — 1 to 50 characters. Non-whitespace characters. Passwords are always case-sensitive.
- **Email** — Up to 100 characters.
- **Profile Picture** — At least 200px X 200px.

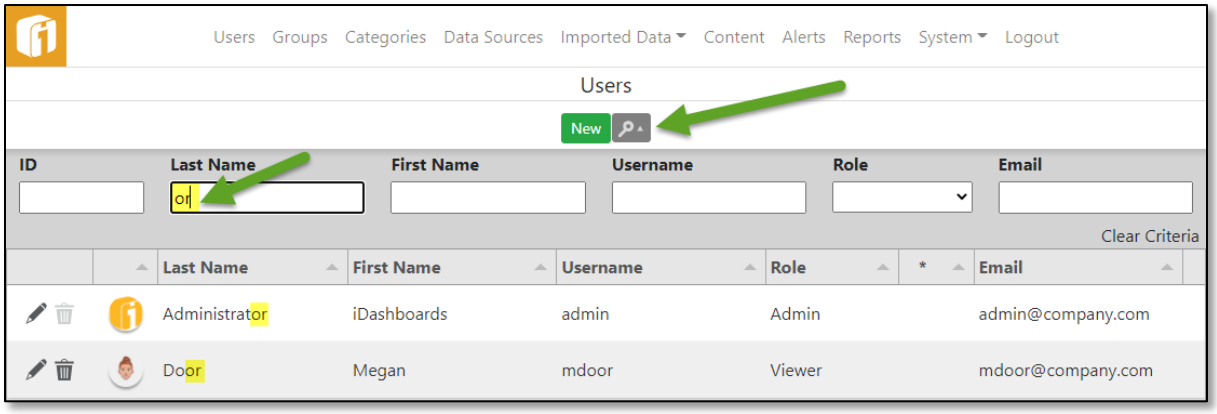
Note: The requirement of a user's email address is controlled by System Security Setting, "Require User Email". See section 13.2.3 , "Security Settings"

7.3 Delete or Modify a User

Any user account except the iDashboards system user can be deleted. To delete a user account, select its Delete icon ( ). To modify a user account, select its Edit icon ( ). The Edit User screen will appear through which the user's basic information can be edited.

To change a users' password, first edit the user account and then check the box for "Change Password". Note that a user's password will NOT be changed unless values are entered into both the "Password" and the "Confirm Password" textboxes, the values match, and they are a legal password (4-50 non-whitespace characters.)

To locate a user account, click the search icon to show the various fields in which to search against. Note that in certain cases of troubleshooting, searching for a user ID can be performed here.



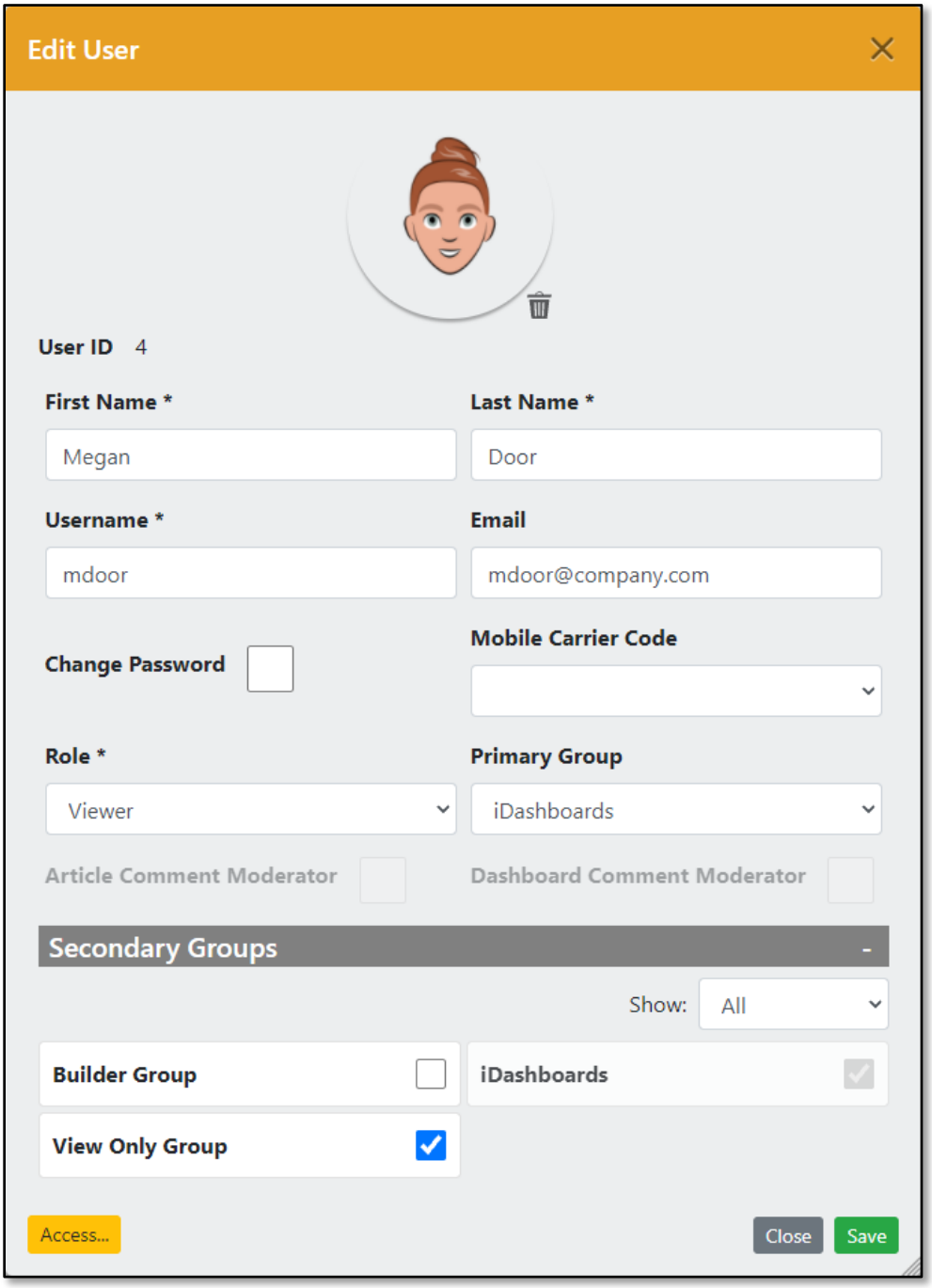
7.4 Understanding Secondary Groups

When a user account is created, it is assigned to a user group. This group is referred to as the user's *primary group*. The primary group of a user can be changed on the edit user screen.

In addition to the primary group, a user may belong to one or more secondary groups. There is no functional difference between the primary and secondary groups. A user's Category access privileges are the combined Category access privileges of all of his or her groups (primary or secondary). If a user has access to a Category through more than one group, the user's access level (view or save) to that Category will be the *maximum* access level provided by his or her groups. Save access is considered "greater than" view access for determining the maximum access level, so if any one of a user's groups has save access to a given Category, the user will have save access to that Category.

7.5 Modifying a User's Secondary Groups

A user's secondary groups are selected in the Secondary Groups section of the Create/Edit User screen. The Secondary Groups screen is accessed by selecting the "Secondary Groups" title on the Create/Edit User screen.

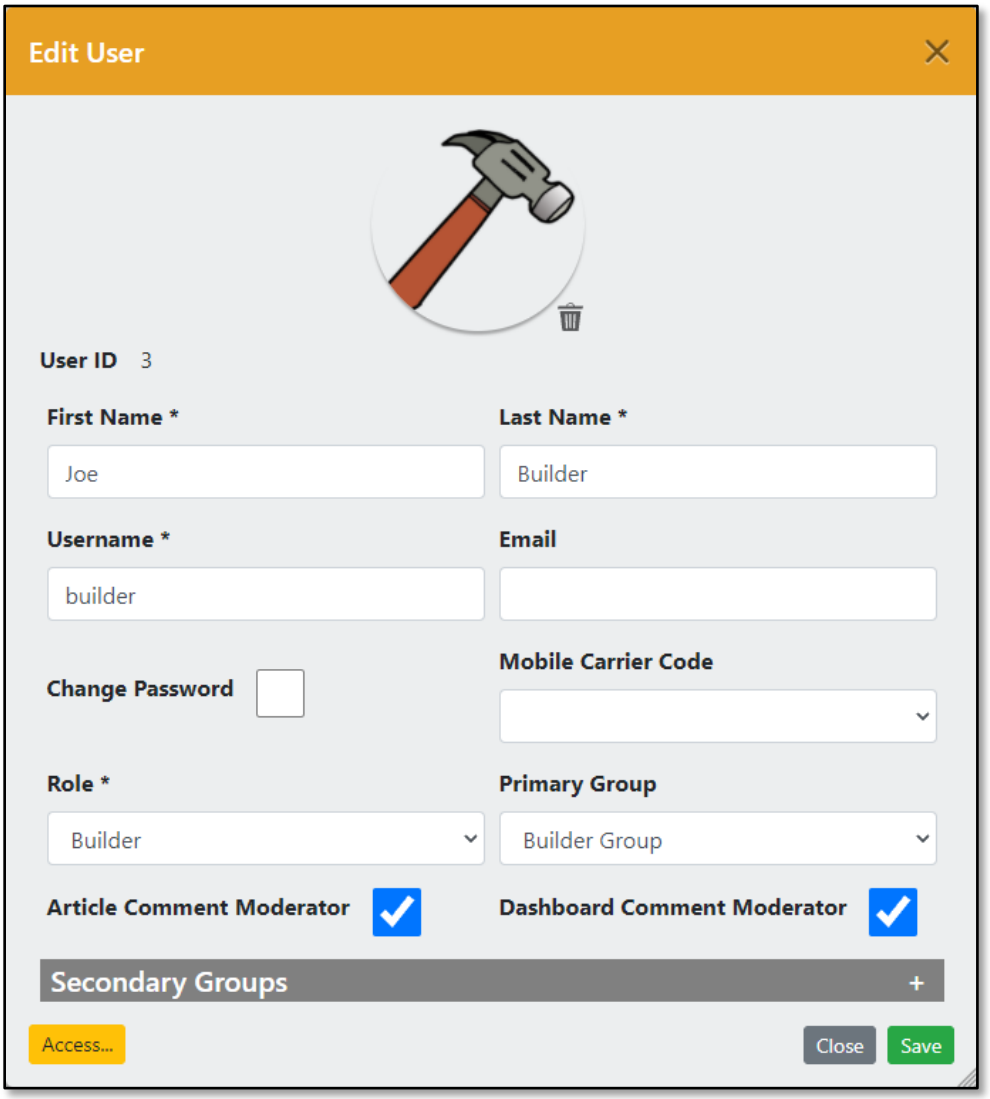


The secondary groups screen lists all of the groups that have been created, with the user's primary group already selected. The user's membership in secondary groups can be

indicated by checking or unchecking the corresponding checkboxes. Select the "Save" button to save the changes, or the "Cancel" button to discard the changes.

7.6 Comment Moderator Control

Only users with a 'Builder' role and above can be moderators. All 'Admin' role users are automatically moderators; others need to be identified as moderators here.



Note: The Knowledge Base feature must be enabled within the iDashboards license. The administrator has the option to disable the entire Knowledge Base feature.

7.7 Access

Selecting 'Access…' will show what Dashboards, Charts, Picklists and Forms are accessible to the user. The access is based on the user's group permissions. Categories with save permission will show  . Changes to the role or groups must be saved before using this.

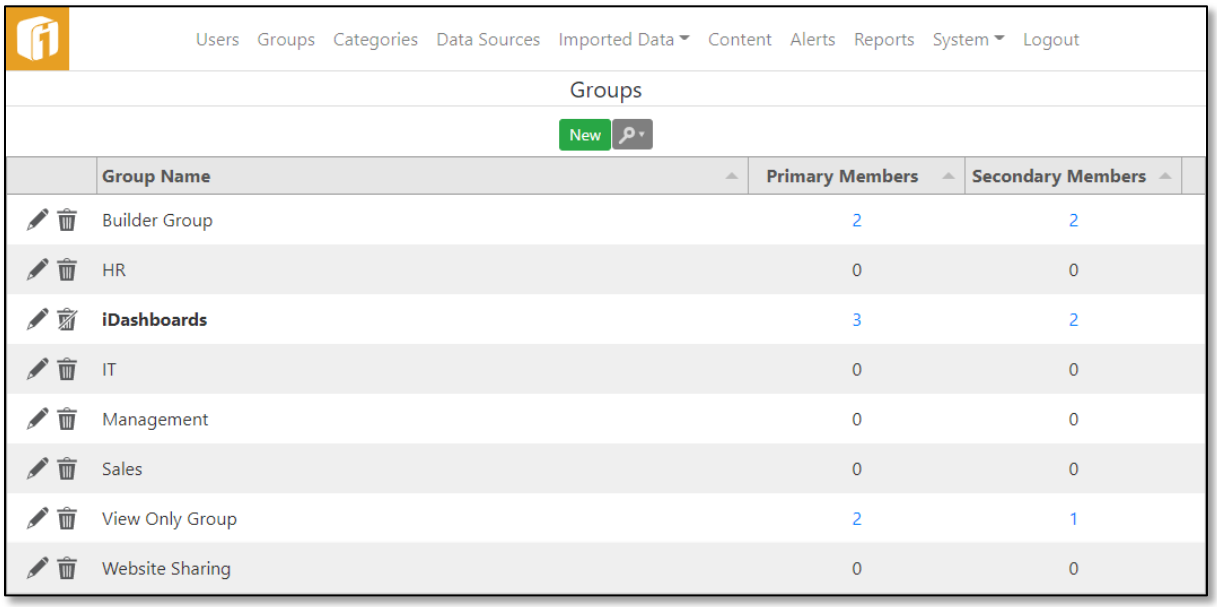
8. Managing Groups

Each iDashboards user must belong to at least one *group* of users. A user's access to charts, dashboards, and other parts of the iDashboards system is determined by the privileges granted to the groups to which he or she belongs. Therefore, assigning a user to a group is the same as assigning a set of access rights to that user.

Typically, the users belonging to a group will share some common characteristic, for example, all users who are shift managers might be assigned to a group called "Shift Managers."

The starting point for managing groups is the Groups screen, which is accessed by selecting the "GROUPS" tile from the Administrator home screen, or "Groups" on the menu bar.

Since iDashboards must always have at least one user group to function properly, a group called "iDashboards" is created as part of the installation process. This is known as the "system group", and it can never be deleted. It can be renamed, however, and otherwise treated like any other group. On the groups screen, the system group will be displayed with a different font color in the list of groups. Any attempts to remove it will fail with a warning message.



8.1 Adding a Group

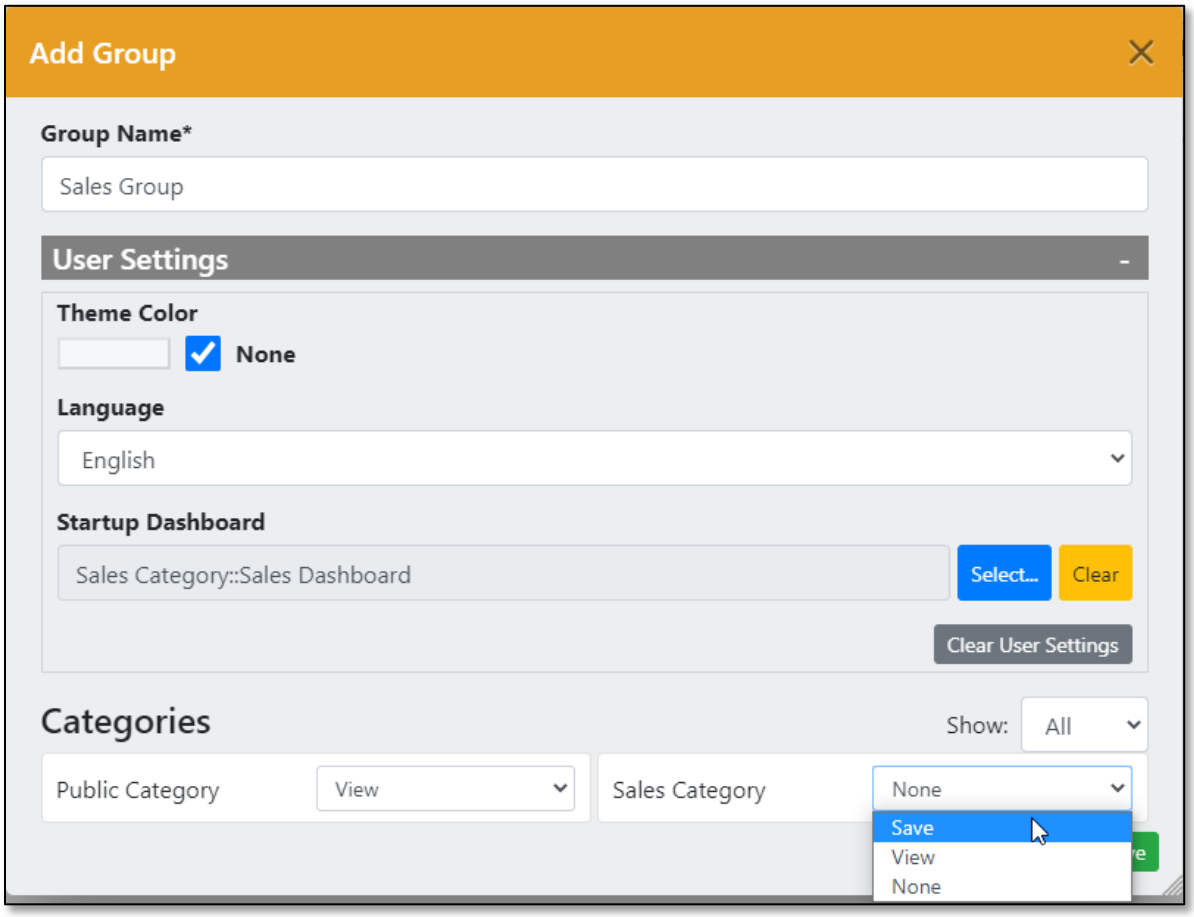
To create a new group, from the Groups Screen, select the "New" button, for the Add Group dialog, and then enter the group's name. A group name may contain no more than 20 characters, and must be unique within iDashboards. Whether or not the uniqueness is case-sensitive depends on the type of database used as the iDashboards repository. For example, Oracle databases are generally case-sensitive, so "Marketing" and "MARKETING" would be considered different names, and could both be used simultaneously as group

names. Microsoft SQL Server, on the other hand, is generally case-insensitive, which means that "Marketing" and "MARKETING" could not both be used simultaneously.

Group level user settings for theme color, language and a startup dashboard are available. When configured they are applied to all users who have it assigned as their primary group.

> *Note: A user can override the group level user settings, using their own user settings and using the Viewer's "Set as Startup Dashboard".*
>
> *Note: An administrator can override all the personal settings, of all users in the primary group, using the* Clear User Settings *button.*

Once the group name has been entered, the group's Category access privileges can be set for each dashboard Category.



The following privilege levels can be selected for each displayed Category:

- **View** means users in the group may view, but not save or delete dashboards or charts in that Category.
- **Save** means users in the group may view, save and delete dashboards or charts in that Category.

- **None** (the blank option) means users in the group have no access to dashboards or charts in a particular Category.

Since all users in a group have "Save" access to their Personal Category it will not appear on under Categories.

Once the group's Category access privileges have been set, select the "Save" button to save the new Group, and Category access privileges, in the repository. Selecting the "Cancel" button will not save anything.

8.1.1 Data Source Access Control

When Data Source Access Control is enabled you can limit the data sources that users within a group may access when building charts. This is controlled under Data Sources.

> *Note: This feature is enabled or disabled through a system setting called "Data Source Access Control Enabled." See Section 13.2.3, "Security Settings", for more information.*
>
> *Note: Chapter 10, Managing Data Sources for more information on data sources.*

8.2 Modifying a Group

To modify the name of an existing group, and its category access privileges, select its Edit icon () on the Group screen. The Edit Group dialog works the same as the Add Group dialog, describe under Section 8.1, "Adding a Group".

8.3 Deleting a Group

Groups other than the system group can be deleted, provided they contain no primary members. A group's primary members are users that have that group as their primary group. To delete a group, find it in the list of groups on the Group screen, verify that its number in the "Primary Members" column is 0, and select its Delete icon (). Then select "OK" on the confirmation dialog that appears.

A group may also have secondary members, which are members that have that group as one of their secondary groups. A group with secondary members but no primary members can be deleted. After it has been deleted, it will no longer appear as a secondary group for any of its former secondary members.

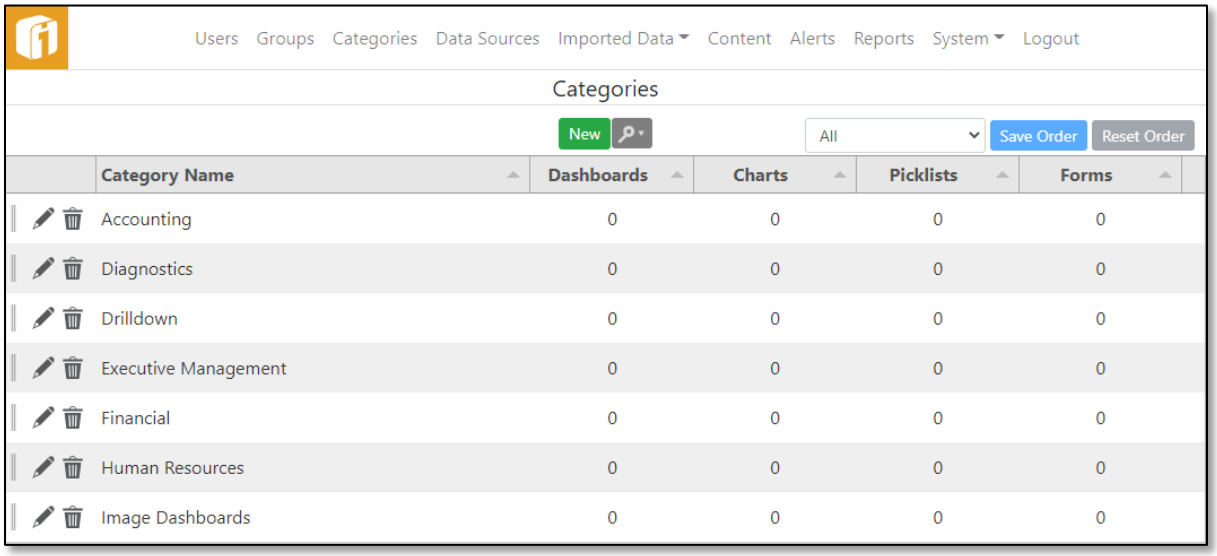
9. Managing Categories

In iDashboards, a Category is what groups logically-related charts and dashboards together. Each chart or dashboard must belong to one and only one Category. Typical categories might be "Sales", "Finance", or "Production." An iDashboards administrator can create and modify categories. A Category can only be deleted if it contains no dashboards, charts, or picklists.

There is a special built-in Category which cannot be created, modified or deleted by an iDashboards administrator, called the "Personal" Category. The Personal Category can be thought of as an individual user's own private workspace. Dashboards saved in the Personal Category are only viewable or accessible by the user who created them. Charts saved in a user's Personal Category are visible to other users only if they are placed in a dashboard that is *not* in the Personal Category.

The starting point for managing categories is the Categories screen. To display it, select the "CATEGORIES" tile from the Administrator home screen, or "Categories" on the menu bar.

Because the "Personal" Category is virtual it will not show on the Categories screen.



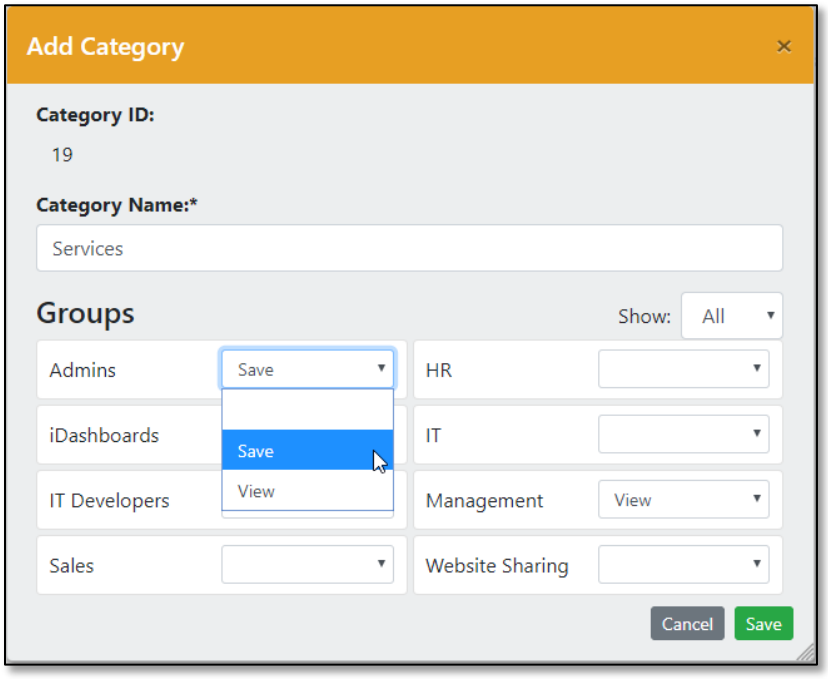
9.1 Adding a Category

Adding a new Category is simple. On the main Categories screen, select the "New" button.

The maximum length for a Category name is 50 characters. To conserve display space Category names should be kept as short as possible while still being meaningful to end users. A Category name must also be unique within iDashboards. Whether or not the uniqueness is case-sensitive depends on the type of database used as the iDashboards repository. For example, Oracle databases are generally case-sensitive, so "Marketing" and "MARKETING" would be considered different names, and could both be used

simultaneously as Category names. Microsoft SQL Server, on the other hand, is generally case-insensitive, which means that "Marketing" and "MARKETING" could not both be used simultaneously.

After the Category name has been entered, group privileges can be set for the new Category.



The available privilege levels are "None", "Save" and "View" (the "None" level is set by leaving the selector blank for a particular group).

After the group privileges have been set, selecting the "Save" button will save the Category and group privileges to the repository. Selecting the "Cancel" button will not save anything.

9.2 Modifying a Category

To modify the name of an existing category, and its group access privileges, select its Edit icon ( ) on the Categories screen. The Edit Category screen works the same as the Add Category screen, describe under 9.1 "Adding a Category".

.

9.3 Deleting a Category

A Category cannot be removed if there are dashboards, charts, or picklists in it. For each listed Category, the number of linked dashboards, charts and picklists are listed in corresponding columns. To remove an existing Category, verify that these numbers are 0, select its Delete icon ( ), and then select "OK" on the confirmation dialog that appears.

9.3.1 Linked Dashboards, Charts and Picklists

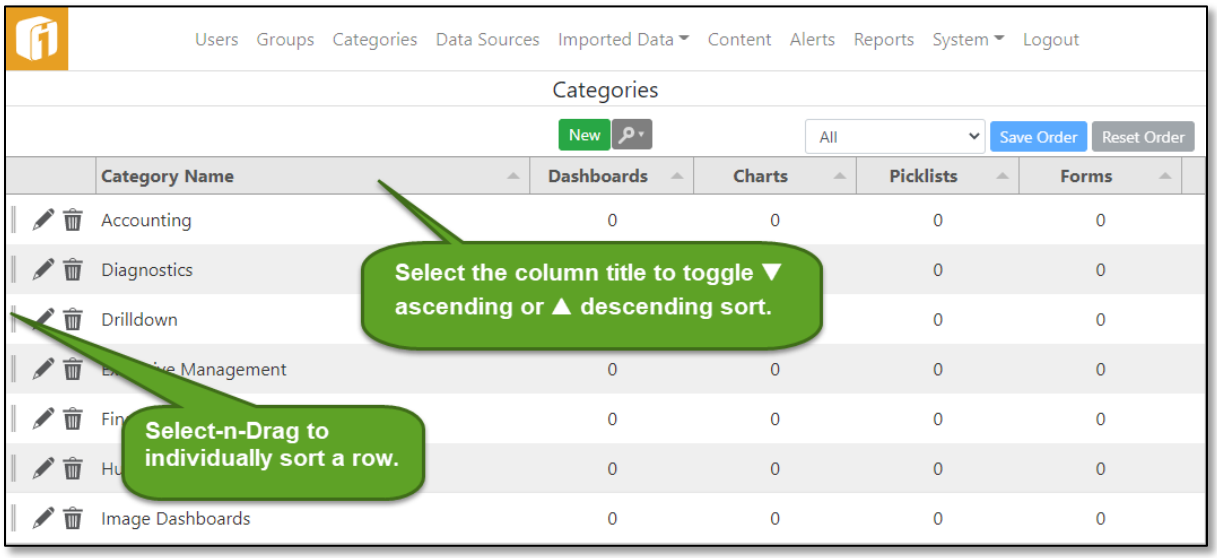
Each Category has a column that contains links to their associated dashboards, charts and picklist. These links not only tell the administrator the number of dashboards, charts and picklists using the Category, but will link to their corresponding list. This feature will assist the administrator attempting to delete Categories that contain dashboards, charts and picklists.

Select the link showing the number of Dashboards, Charts or Picklists for a Category to display a dialog listing the details for that object type. From this screen, you can delete any dashboard, chart or picklist. A new screen displaying the associated dashboards, charts or picklists will load. From this screen, you can delete any dashboard, chart or picklist.

> *Note: If the Forms feature is enabled, within the iDashboards license, a column for linked Forms is also included.*

9.4 Sorting Categories

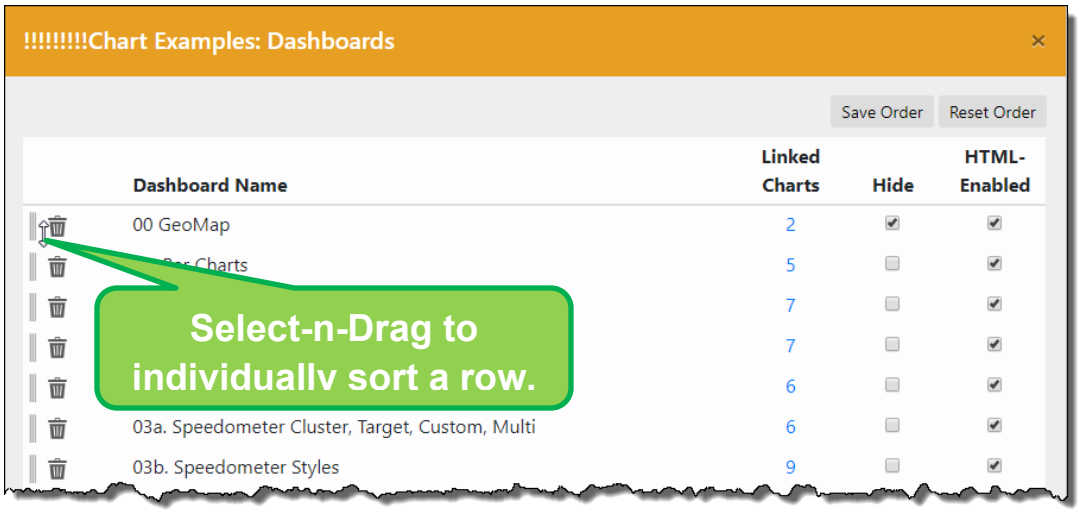
Categories can be sorted, by the administrator, in any order. There is the ability to alphabetically sort all Categories by selecting the column title and toggling ▼ ascending or ▲ descending. To individually sort categories, Select-n-Drag the category using the reorder icon ( ) at the beginning of each category row. To keeps this sort order select the "Save Order" button. To go back to the original sort order select the "Reset Order" button.



Select the column title to toggle ▼ ascending or ▲ descending sort.

9.5 Sorting Dashboards

Dashboards within a Category can be sorted, by the administrator, in any order. To individually sort dashboards, Select-n-Drag the Dashboard using the reorder icon (|||) at the beginning of each dashboard row. To keeps this sort order select the "Save Order" button. To go back to the original sort order select the "Reset Order" button.



9.5.1 Hiding Dashboards

There is an option to choose if a dashboard should appear in the category list or if the dashboard should be hidden. Dashboards with the "Hide" checkbox selected will not appear under that Category in the iDashboards View interface.

Hiding a Dashboard within a Category is especially beneficial when drilling down into a dashboard. Hiding the target dashboard will keep users from opening the dashboard natively.

> *Note: This option is also available through the iDashboards Build application within the Extended Dashboard Properties.*

There is also the option to choose if a dashboard should be viewed in HTML. Not all dashboards will render HTML and can be hidden to avoid visual quality issues.

9.6 Protected Categories

A lock icon indicates the content saved within the category has been authored, and protected, by iDashboards. Contact support for more information.

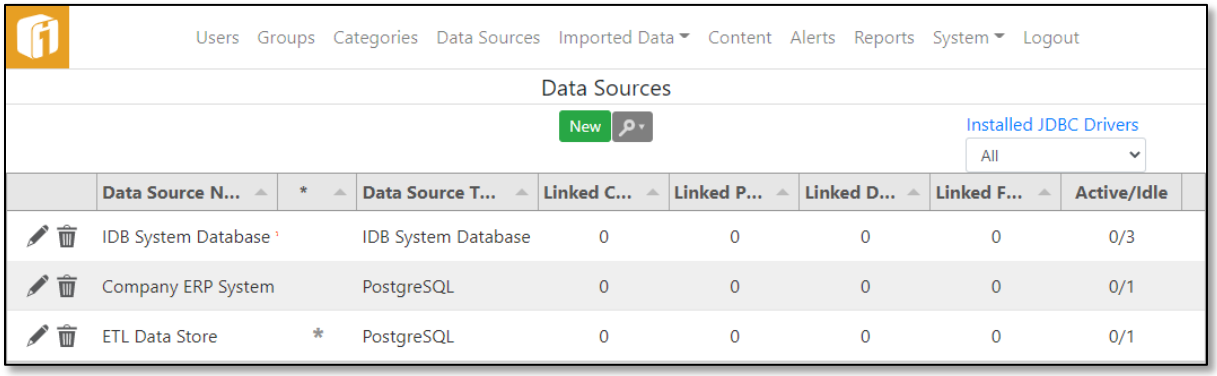
10. Managing Data Sources

In iDashboards, charts may be built from "dynamic data", i.e., data read directly from objects such as tables or views in a production database. These objects may be in the same database as the iDashboards repository tables, or they may in any number of other external databases, such as a data warehouse. In order to use a database's tables or views for dynamic chart data, the database must be configured as a "Data Source" in iDashboards. This applies even to the iDashboards repository database, although it is automatically configured as a Data Source during the iDashboards installation process.

Configuring a Data Source usually requires special knowledge about the target database, for example usernames, passwords, port numbers, host names, etc., so it generally should be done by, or with the assistance of, a database administrator (DBA).

Data Sources are managed through the Data Sources screen, which is accessed by selecting select the "DATA SOURCES" tile from the Administrator home screen, or "Data Sources" on the menu bar.

The Data Sources screen displays a list of currently configured data sources. If one of the listed Data Sources is the iDashboards repository database, it will be marked with a red asterisk (*) after its name. The default name it is given during the iDashboards installation process is "IDB System Database", although this can be changed.



10.1 Understanding JDBC Drivers

One of the more powerful features of iDashboards is its ability to connect simultaneously to multiple databases from different vendors. This is accomplished through JDBC technology. JDBC, which stands for *Java Database Connectivity*, is an API that provides a standard, uniform interface through which an application can interact with any JDBC-enabled database. A JDBC-enabled database is one for which a JDBC "driver" is available. Today, JDBC drivers are available for virtually every enterprise-class database. In most cases, they are bundled directly with the database product or can be downloaded for free from the database vendor's website.

10.1.1 Installing JDBC Drivers

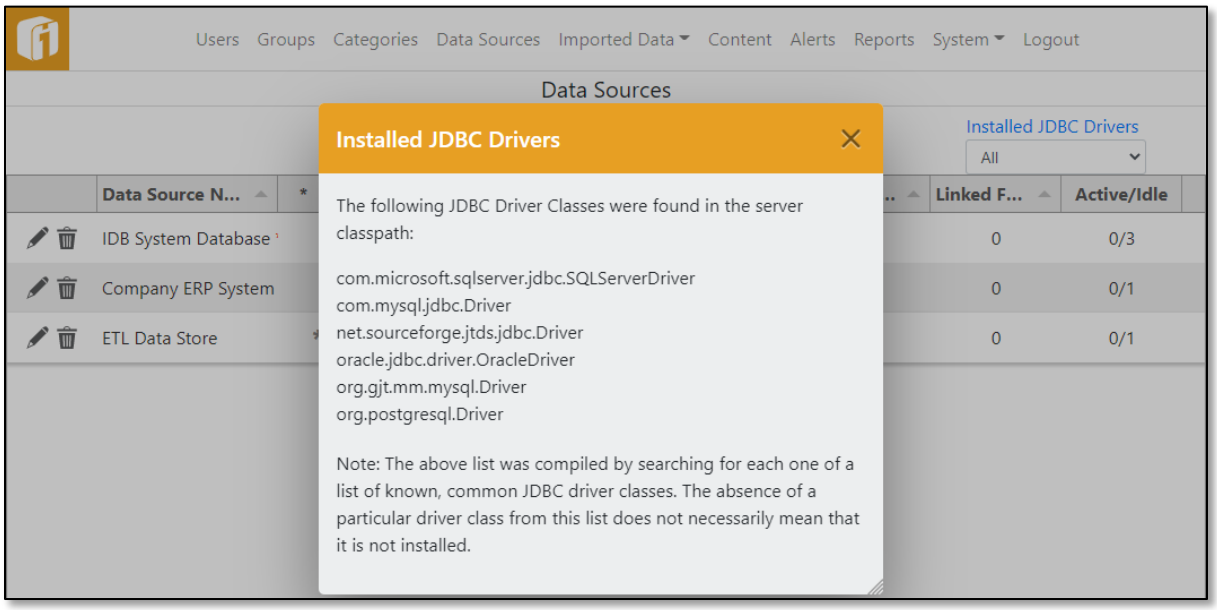
A JDBC driver is usually packaged in a single file with a ".jar" filename extension. (In the Java world, this type of file is referred to simply as a "jar file.") For a driver to be used by iDashboards, its jar file must be placed in the <IVIZGROUP HOME>\drivers directory, or in the classpath of the application server.

10.1.2 Verifying JDBC Driver Installation

Each JDBC driver has a unique "driver class name" used to identify it and load it into memory at runtime. The driver class name consists of several words separated by periods, for example, Oracle's JDBC driver class name is *oracle.jdbc.driver.OracleDriver*.

iDashboards checks for the presence of a number of common JDBC drivers. Select the "Installed JDBC Driver" link, at the top right corner of the Data Sources screen, to display a list of the ones that are installed and available. In this example eight different JDBC drivers have been detected.

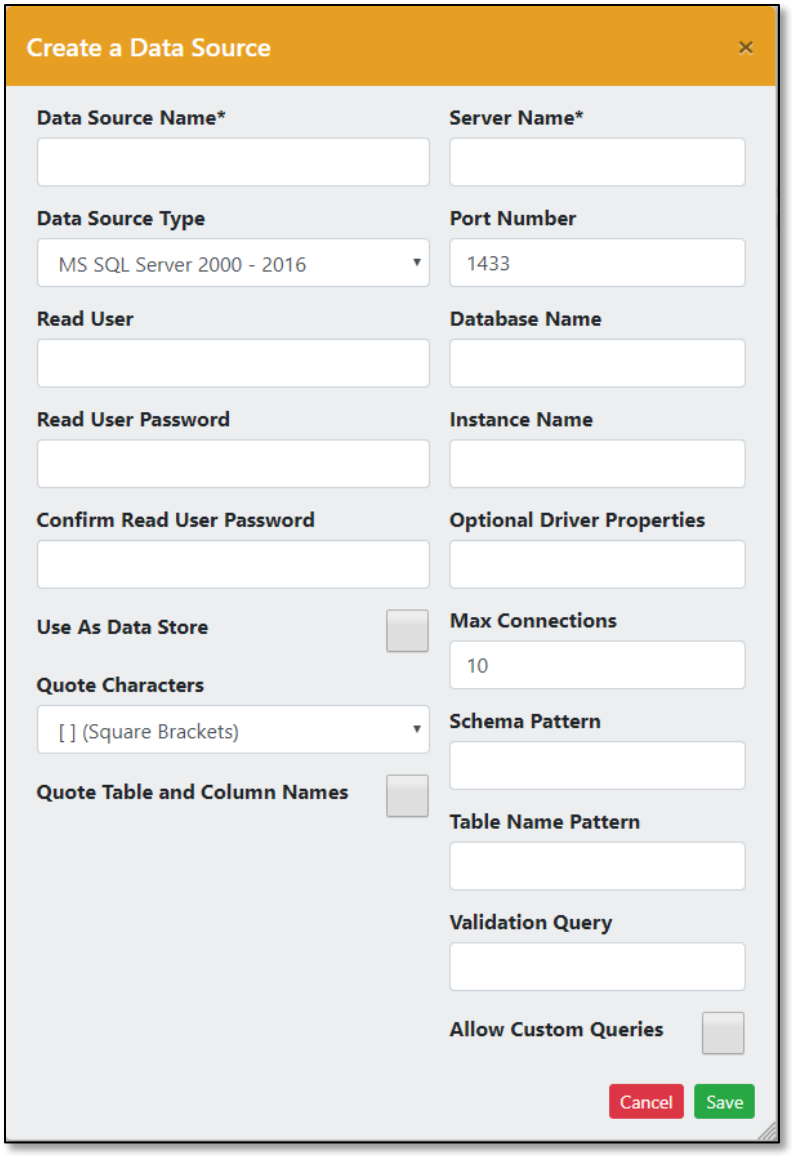
In order to connect to any external data source, it is essential that the proper JDBC driver is installed. If difficulty is encountered contact iDashboards technical support for assistance.



10.2 Adding a Data Source

Once the required JDBC drivers have been installed, a Data Source may be configured. To add a Data Source select the "New" button to launch the "Create a Data Source" screen.

Selecting different Data Source Types may result in varying fields compared to the image below. If the field is required by iDashboards, it will be identified with an asterisk. However, specific data sources may have additional field requirements not designated with an asterisk.



- **Data Source Name** — This is an arbitrarily-chosen name used to identify this Data Source. It can be from 1 to 60 characters.
- **Data Source Type** — Select the type of Data Source you will be adding from the dropdown list. The Data Source type may be specific to the type of database that is being connected to, or it may be a generic type such as "Generic JDBC". Just because a particular database is not listed in the dropdown does not mean it is not supported by iDashboards; any database for which JDBC drivers are available may be used. If the specific database type is not listed, select "Generic JDBC" if using a JDBC driver.
- **Server Name** — This can be the database server's name or it's IP address.

- **Port Number** — This is the port number on which the database server receives incoming connections.
- **Database Name** — This is the name of the target database on the server.
- **Instance Name** — The instance of the target database, if needed.
- **Read User** — This is the User ID that iDashboards will use to connect to the target database.
- **Read User Password** — This is the password that iDashboards will use to connect to the target database. It will be stored in encrypted form in the iDashboards repository database.
- **Confirm Read User Password** — If a password is entered, the same password must be entered in the "Confirm Password" textbox.
- **Quote Characters** — This dropdown list contains the options: *" " (Quotation Marks)*, *[ ] (Square Brackets) and ` ` (Back Ticks)*. This is used by iDashboards when building SQL queries. It should only be changed from its default setting if problems are encountered, and with the assistance of iDashboards technical support.
- **Quote Table and Column Names** — When this option is checked, the SQL that is generated for a table- or view-based chart will have the table or view name, and all of the column names, enclosed in the quote characters for that data source. This option will allow for the use of reserved words as table and column names in data sources.
- **Optional Driver Properties** — Some JDBC drivers accept driver-specific properties that can control various aspects of the driver's behavior. In virtually all cases, an entry should not be required for this property; however, database I/O can sometimes be optimized through the use of these parameters. Each property should be entered in the form:

  ```
  propertyname=propertyvalue
  ```

  and multiple properties should be separated with semicolons, for example:

  ```
  property1name=property1value;property2name=property2value
  ```

- **Max Connections** — iDashboards creates and maintains a "pool" of connections to the target database for each configured Data Source. Pooling connections improves performance while minimizing resource consumption. The "Max Connections" parameter is an integer which indicates the maximum size to which the connection pool will be allowed to grow. (The initial pool size is 3, and it only grows if needed.) Normally, this should be left at its default value of 10. It is unlikely that the iDashboards server will ever be busy enough to require a larger connection pool.

In certain cases, it is preferable *not* to maintain a pool of open connections, because it locks the database file, preventing access by other programs. In such cases, a "non-pooled" Data Source can be created by setting the "Max Connections" parameter to 0. Each time iDashboards needs to retrieve data from a non-pooled Data Source, it connects, reads the data, and then disconnects.

- **Schema Pattern** — This can be used to filter the list of schemas from the target database that will be available for dynamic chart data. The term "schema" is often used interchangeably with "table owner" or "database user." If this field is left blank, then all of the schemas visible to the user named in the "UserID" field will be available. To further filter this list, a "pattern" can be applied, which may include the SQL wildcard characters "%" (matches any string of zero or more characters) or "_" (matches any single character.) The pattern may or may not be case-sensitive, depending on the target database.

- **Table Name Pattern** — This can be a pattern similar to a schema pattern that will filter the list of tables available for dynamic chart data. If it is left blank, then all of the tables for all of the available schemas will be made available for dynamic chart data.

- **Allow Custom Queries** — If this box is checked, users with the Builder role can build charts based on custom queries, which are free-form SQL statements executed in the target database.

The fields between the first five and the last six are specific to the Data Source type being created. In the case of database-specific types, the names of these fields should be self-explanatory and the required values should be known by the administrator of the target database. The Generic JDBC types are special cases, however, since they can be used with virtually any type of relational database. The type-specific fields found on their modification forms are explained below.

- **Validation Query** — If a Data Source is known to go offline occasionally, while the iDashboards server remains online, then a value for this property should be provided. It should be a short, simple SQL SELECT statement that is valid for the data source and will always return at least one row. Whenever a connection is fetched from the connection pool, this query will be run to determine if the connection is still active. If the query fails, the connection will be considered "dead" and removed from the pool. Because this query will be run often, it should be one that is known to run quickly to minimize the performance cost. Oracle databases often include a special system table named "dual" that is ideal for this purpose; in such cases, the statement "SELECT * FROM DUAL" may be used.

Generic JDBC — If "Generic JDBC" is selected as the Data Source type, the following fields should be completed (in addition to the standard ones listed above) on the Data Source edit form:

- **Database URL** — This is a string that contains all of the information the JDBC driver needs to connect to the target database. It usually contains information such as the hostname, database name, and possibly the port number on which the database server receives incoming connections. The format of the database URL depends solely on the JDBC driver being used, and can vary widely from one driver to the next. The required format should be explained in the driver's documentation, however, Section 3.4.8, "JDBC Driver Configurations" contains sample URLs for a number of popular JDBC drivers.

- **JDBC Driver Class** — This is the "class name" for the JDBC driver being used. As explained above, the driver class name consists of several words separated by periods, and is used to identify and load the JDBC driver at runtime. The class name should be provided in the driver's documentation, however, Section 3.4.8, "JDBC Driver Configurations" lists the class names for a number of popular JDBC drivers.

Once the form fields for the new Data Source have been completed, select the "Save" button to save the changes. iDashboards will attempt to connect to the target database using the supplied parameters. If it fails to connect to the target database, an error message will be displayed (which may contain diagnostic information) and the form will remain displayed so corrections can be made. If it connects successfully to the target database, the new Data Source will be created in the iDashboards repository.

When the new Data Source has been created, the next screen displayed will depend on whether or not Data Source access control is enabled. If it is disabled, the main Data Sources screen will be displayed. If it is enabled, the Data Source's group access screen will be displayed. See Section 10.6, "Data Source Access Control" for more information on Data Source access control.

10.2.1 Use as Data Store Property

For an iDashboards Data Hub application's ETL job to "write" into a database table, the Data Source needs to be configured as a Data Store. Enabling this option will require a "Write User" and "Write User Password" for securely controlling this permission. Only the supported databases listed below can be flagged as a Data Store:

- Microsoft SQL Server (2005 or later)
- Oracle (9i or later)
- MySQL (5.0.3 or later)
- PostgreSQL (9 or later)

10.3 Adding the Repository Database as a Data Source

As mentioned previously, the iDashboards repository database is configured as an available Data Source during the iDashboards installation process, and it can also be removed from the list of available Data Sources. If it is not in the list, a button will appear, stating "Add iDashboards System Database". Simply select this button to make the repository database available as a Data Source.

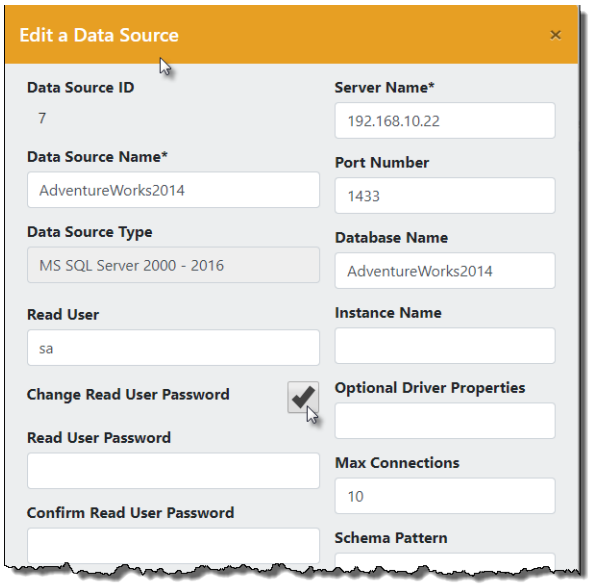
10.4 Modifying a Data Source

To modify an existing Data Source's connection properties, select its Edit icon ( ). The "Edit a Data Source" dialog will appear, which contains a form similar to the one originally used to configure the data source. Any of the values can be changed, however care must be taken that changes will not "break" charts that rely on that Data Source. For example, all of the Data Source's tables, columns, schemas or stored procedures that have been used to build charts must still be available after the changes have been made.

Once the changes have been made, select the "Save" button to save the changes, or "Cancel" to discard the changes and return to the Data Sources screen. After the "Save" button has been selected, iDashboards will attempt to connect to the target database using the new parameters. If it succeeds, the old connection pool will be discarded, a new one will be created, and a success message will be displayed. If it fails, the existing connection pool will be preserved, an error message will be displayed (which may contain diagnostic information) and the dialog will remain displayed so corrections can be made.

10.4.1 Modifying Data Source Password

Sometimes, the data source login credentials need to change. First, find and edit the data source. Then, check the box to "Change Password". Once this checkbox is checked, then the "Password" and "Confirm Password" fields must match. If it is not checked, no password modifications will occur. If the checkbox is checked and both the "Password" and "Confirm Password" fields are blank, then any existing password will be removed from the Data Source's configuration, and it will have a blank password.



10.5 Removing a Data Source

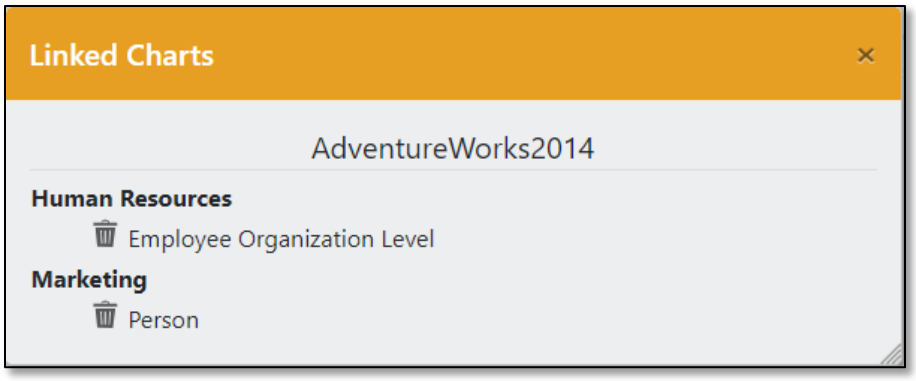
A Data Source cannot be removed if there are any charts, picklists and data sets linked to it (i.e. they pull their data from it.) For each listed Data Source, the number of these linked

items appear in columns next to it. To remove an existing Data Source, verify that this numbers are all 0, select its Delete icon ( ), and then select "OK" on the confirmation dialog that appears.

10.5.1 Linked Charts, Picklists and Data Sets

Each Data Source has a column that contains links to their associated charts, picklists and data sets. These links display the number of items using the Data Source and link to their corresponding item list. This feature assists the administrator attempting to delete Data Sources that are used by charts, picklists and data sets.

Select a number in the "Linked Charts" column to see the list of charts utilizing the Data Source. If the number is "0" (zero) then there will not be a link available. A dialog displaying the associated charts will appear. The category name will be displayed above the listing the charts. Next to the name of the chart there is a remove button which will delete the chart.



"Linked Picklists" and "Linked Data Sets" work the same way. For each listed Picklist, the number of linked charts and dashboards appear in columns next to it, and selecting the number will again open a dialog that will display a list of their names.

> *Note: If the Forms feature is enabled, within the iDashboards license, a column for linked Forms is also included.*

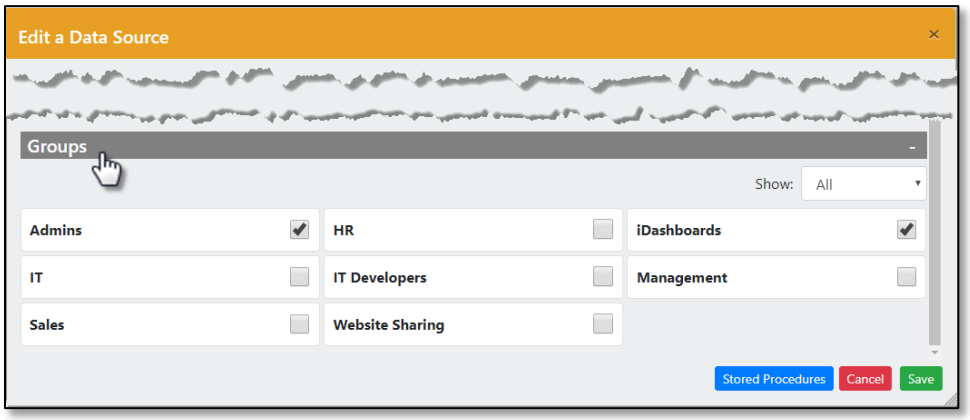
10.6 Data Source Access Control

Data Source Access Control is a feature that can be used to limit the groups that are allowed to access a particular Data Source for the purpose of building charts. The feature, which is disabled by default, can be enabled or disabled through a system setting called "Data Source Access Control Enabled", which is in the "Security Settings" category of system settings. See Section 13.2.3, "Security Settings", for information on how to change a system setting.

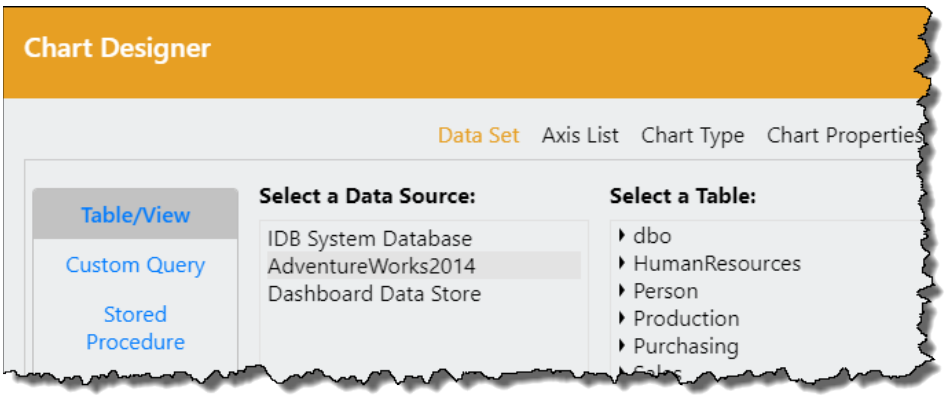
If Data Source access control is disabled, then a user with the Builder role will be able to access *any* configured Data Source when creating a chart in the iDashboards. If it is

enabled, then a group must be granted explicit access to a Data Source before Builders within that group can create a chart based on that Data Source.

When (and only when) the Data Source access control feature is enabled, a Data Source's group access section will be available when a Data Source is added or modified. The group access section will list all of the groups that are configured within the iDashboards system. It can be accessed by selecting a Data Source's Edit icon ( ) on the main Data Sources screen, and then scrolling down to the "Groups" title, and select it to expand that section.



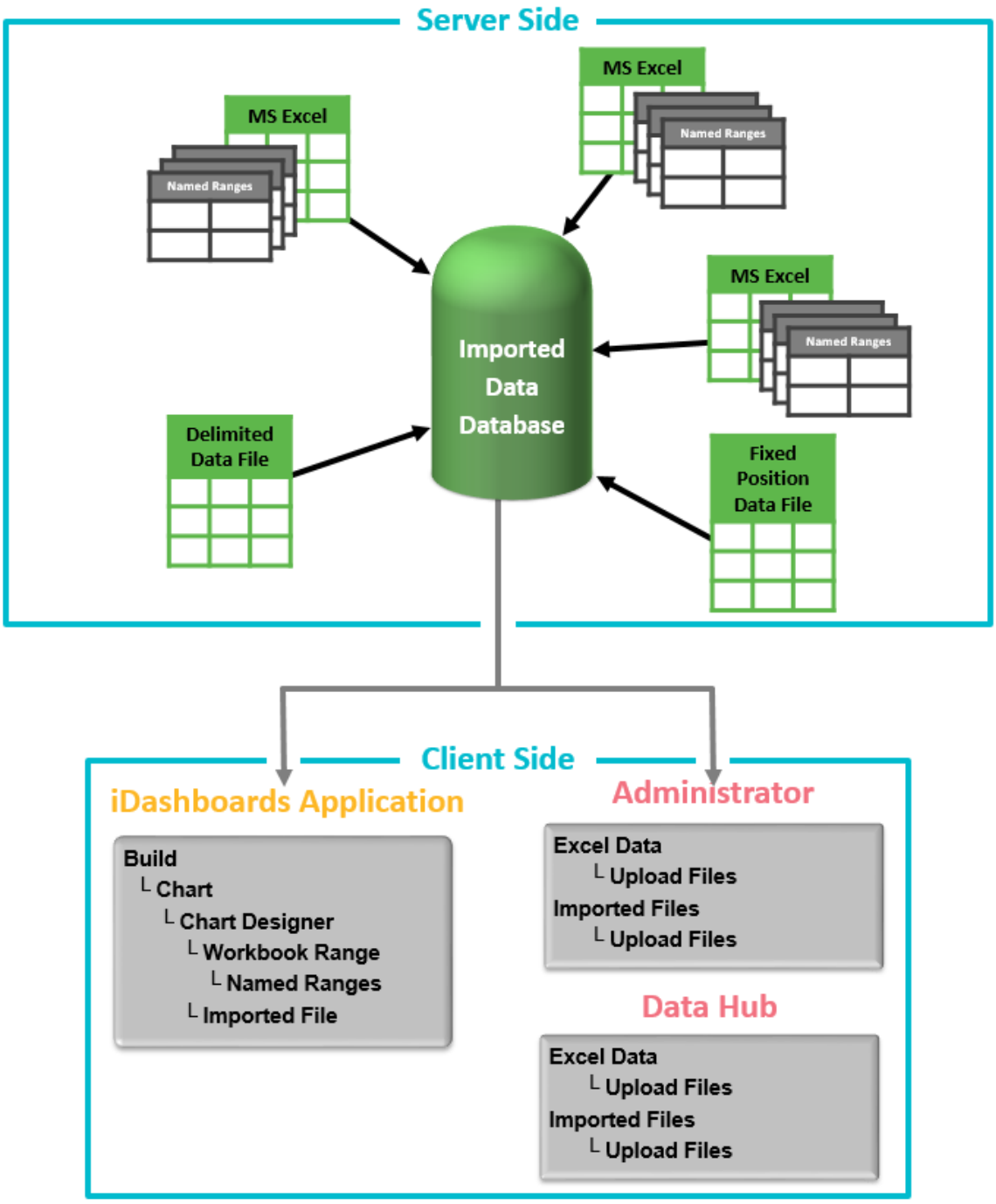
When a checkbox for a particular group is checked on a Data Source's group access screen, users who belong to that group and also have the Builder role will see that Data Source in the list of available Data Sources displayed in the iDashboards Build application in the course of building a chart.



To save changes made to the group access screen, select the "Save" button. Select the "Cancel" button to discard changes and return to the main Data Sources screen.

11. Imported Data Sources

Microsoft Excel, and Delimited and Fixed Position, data files can be equally managed between the iDashboards Administrator application and the iDashboards Data Hub. However, the context of this manual is to describe how to use these files for building charts within iDashboards. All files will be uploaded into the singular Imported Data Database.



11.1 Usage Requirements

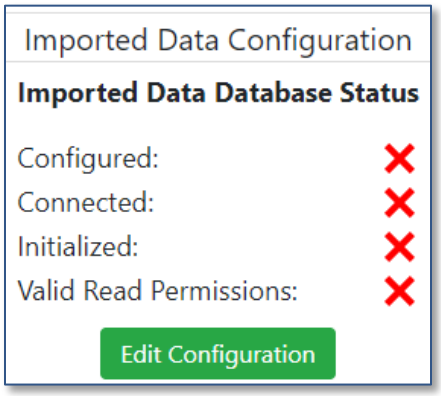
1) Configured Imported Data database
2) **Excel** files
   a) *.xls, *.xlsx files
   b) Must contain named ranges
   c) Macros (*.xlsm) are not supported
3) Data files
   a) **Delimited**
      i) *.csv files
      ii) The Default is comma delimited
      iii) Uses an optional column header row
   b) **Fixed** Position
      i) *.txt files
      ii) Each column is in a specific position in the record
      iii) Does not use a column header row, columns are defined during import
4) Duplicate Column Names are not allowed
5) File size must be less than 10MB

11.2 Configure the Imported Data Database

Before you can begin uploading Excel or Data files, you must first have a configured Imported Data database. The database must first be created on your database server and a user with proper permissions should be available. See Section 3.4.3, "Creating the Imported Data Database" for more information.

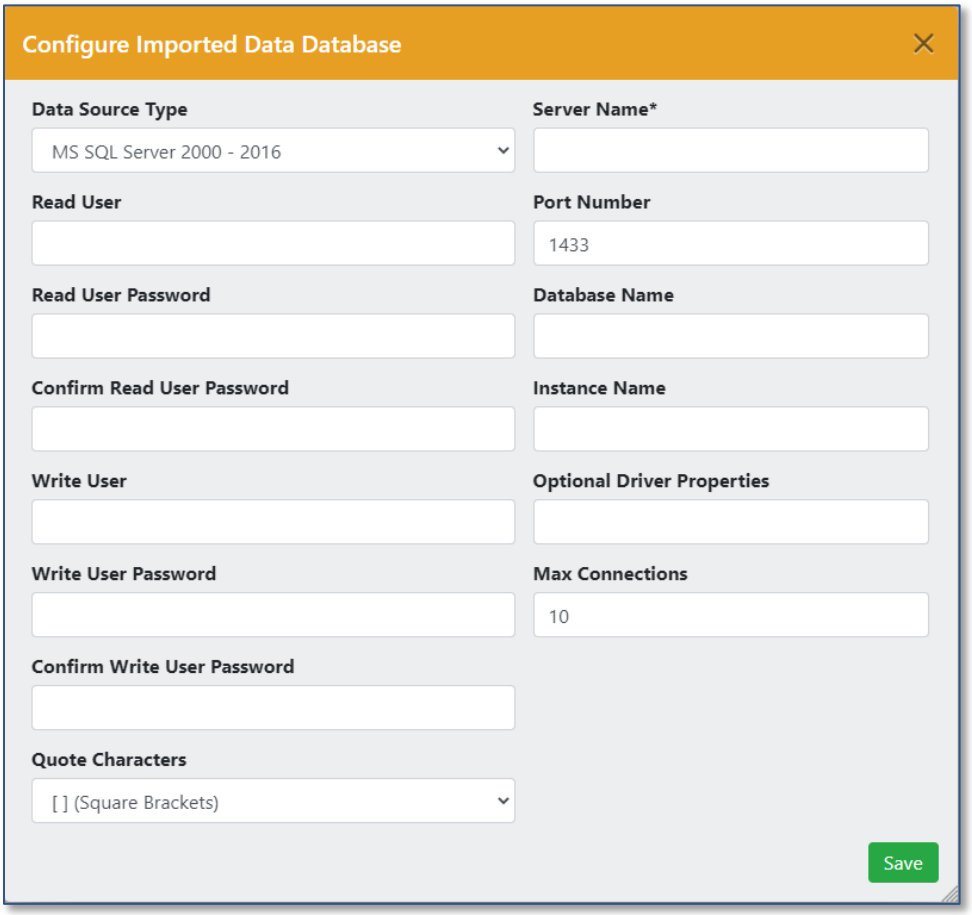
Once you have created the Imported Data database, follow the steps below to configure the database for use.

1. From the System menu, select "Imported Data Configuration" a screen will display the configuration status.

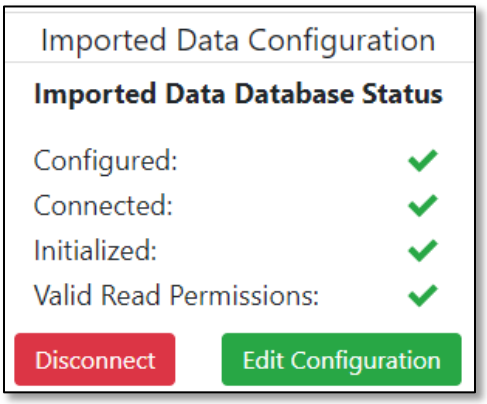


2. Select "Edit Configuration" to open the Configure Imported Data Database. Enter the information required to connect to your newly created database. All fields marked

with an asterisk are required by iDashboards, but your database server may require additional information to make the connection.



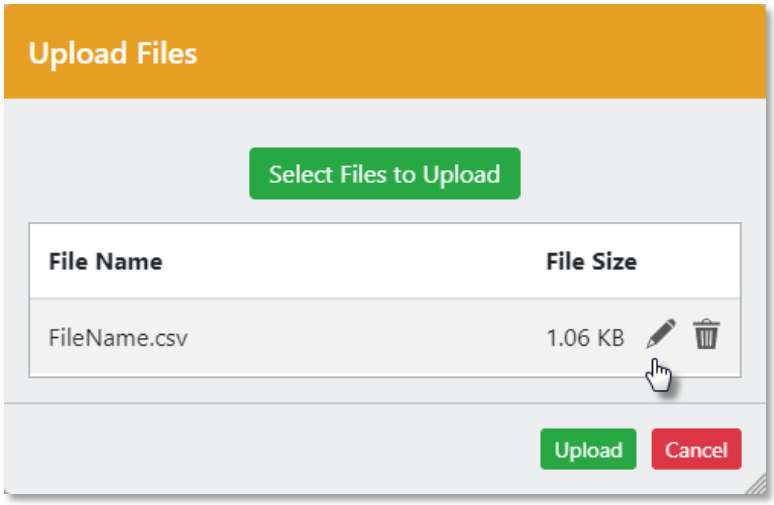
3. Once you have entered your connection information and select "Save".
4. If the database was configured correctly, you should see a green checkmark next to all of the status lines in the Imported Data Database Status screen.



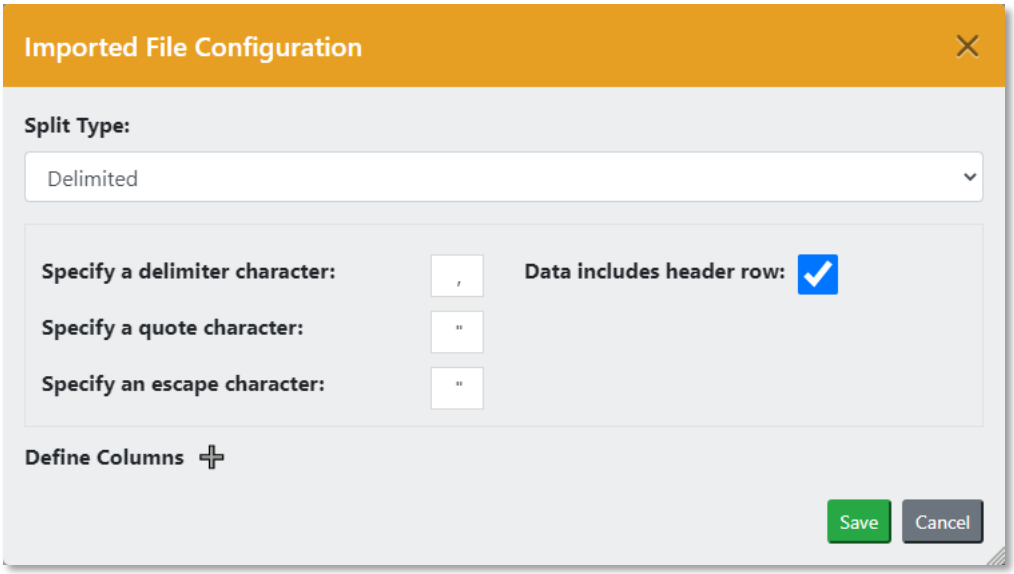
You can now use "Imported Data" to choose "Excel Data" or "Imported File", and then use the "Upload Files" button.

11.3 Imported Files

The primary difference between Excel Data and Imported File is that the administrator can specify how the Imported File is parsed. This is accomplished through the Imported File Configuration screen. It is access by using the pencil icon for the specific file.

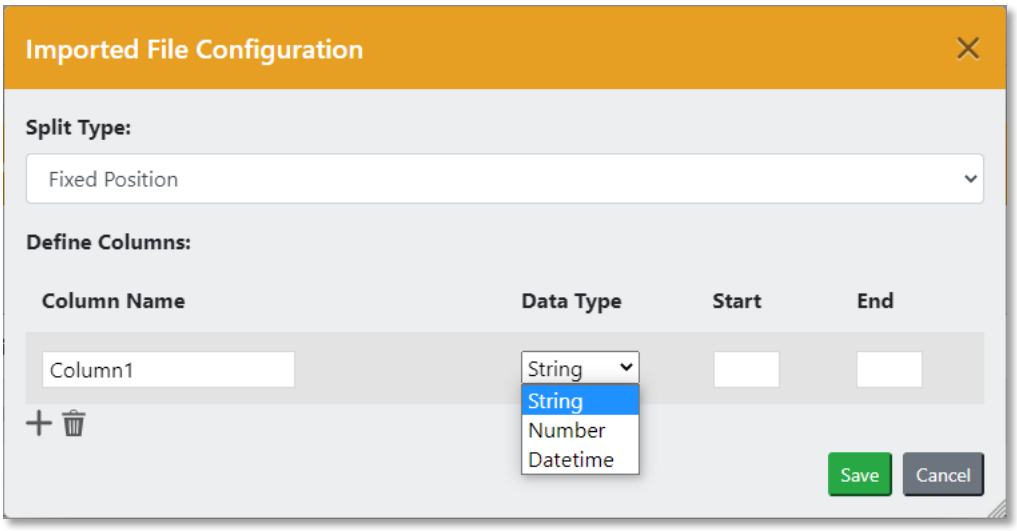


For Delimited Files, user can define the columns, either in the file as a header record, or using the Imported File Configuration. Using this, a delimiter is specify and whether there is a header row. The default file extension is .csv. This is the default configuration:



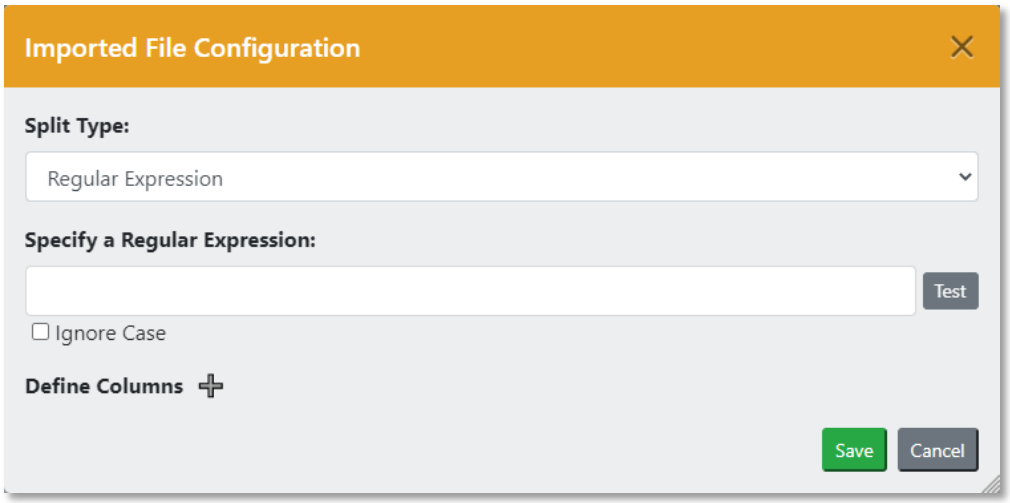
If no columns are defined, the columns names and data types are inferred from the data. This is based on whether there is a header row and the idbdata.text.max.rows.to.scan setting in the ivizgroup property file. The default for this property is -1, which means scan all rows to determine the data type. If columns are defined, they take priority over inferred columns.

For Fixed Position files, all column must be defined using the Imported File Configuration. The default file extension is .txt. This is the default configuration:



For Regular Expression files, using the Imported File Configuration, a regular expression is provided along with capture groups. The Regular Expression pattern does not need to specify the entire string, but it does need to specify a substring, by using non-capture groups if necessary.

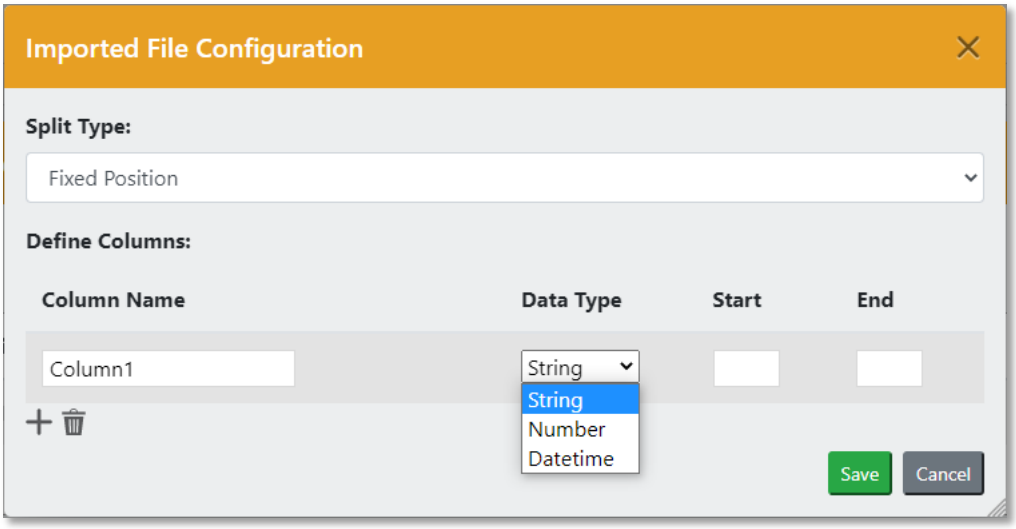
If not provided, the column names will be Column1, Column2, Column3, etc. Data types will be inferred from the data based on the idbdata.text.max.rows.to.scan setting, in the ivizgroup property file. The default for this property is -1, which means scan all rows to determine the data type. If column definitions are provided, the number of columns must match the number of capture groups. This is the default configuration:



If no columns are defined, the columns names and data types are inferred from the data. This is based on whether there is a header row and the idbdata.text.max.rows.to.scan

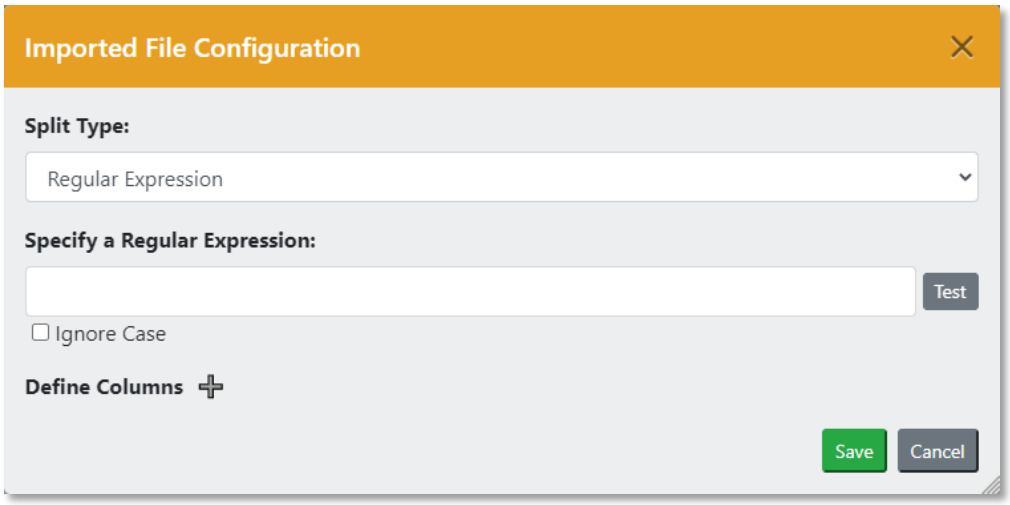
setting in the ivizgroup property file. The default for this property is -1, which means scan all rows to determine the data type. If columns are defined, they take priority over inferred columns.

For Fixed Position files, all column must be defined using the Imported File Configuration. The default file extension is .txt. This is the default configuration:



For Regular Expression files, using the Imported File Configuration, a regular expression is provided along with capture groups. The Regular Expression pattern does not need to specify the entire string, but it does need to specify a substring, by using non-capture groups if necessary.

If not provided, the column names will be Column1, Column2, Column3, etc. Data types will be inferred from the data based on the idbdata.text.max.rows.to.scan setting, in the ivizgroup property file. The default for this property is -1, which means scan all rows to determine the data type. If column definitions are provided, the number of columns must match the number of capture groups. This is the default configuration:



11.3.1 File Name Rules

Under the hood, iDashboards requires uniqueness for the first 11-characters of a file. The first 11-characters are calculated by omitting any spaces and making all letters UPPERCASE.

For end-users, files without the first 11-characters being unique can be uploaded until there are 100 occurrences. When attempting to upload the 101st file (with a naming collision) the user will be notified of this rule and will have to change the filename before the upload will occur. The change needs to be a non-space character change within the first 11-characters.

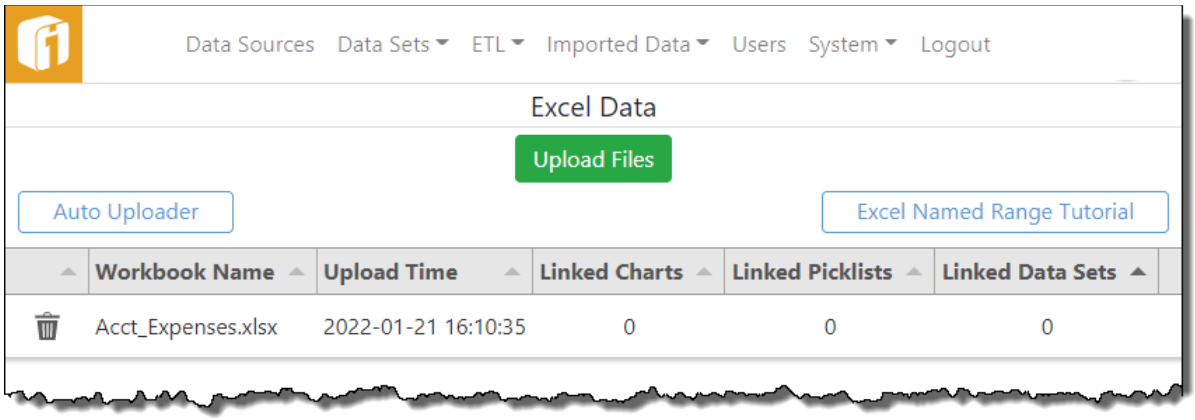
11.4 Automated File Uploads

To automate the process where files need to uploaded, there is an automated file uploader. The iDashboards Auto Uploader allows for direct uploading of files from a client machine to the iDashboards environment without requiring the user to log in to upload their data sources. The Auto Uploader allows files to be uploaded immediately or scheduled for repeated uploads using a scheduling mechanism.

11.4.1 Enable the setting

In the Data Hub, navigate to System > Settings > Server Settings, and then review the option "Imported File Auto Upload Enabled". This setting determines whether or not users can use the Auto Uploader desktop application to upload files to this server application. Set this to TRUE to enable the option.

Then, to access to the iDashboards Auto Uploader and the associated documentation, log into the iDashboards, navigate to "Excel Data" or "Imported Files" and click the button "Auto Uploader" and then follow the instructions.



11.4.2 Auto Uploader System and User Requirements

The IAU has the following minimum system requirements:

- Microsoft Windows 7 or Windows Server 2008 R2
- Microsoft .NET 4 Framework – Full profile

It also requires that the user running the program to have administrator privileges on the PC for which the iDashboards Auto Uploader is installed. These elevated privileges are necessary to create and execute tasks in the Windows operating system.

12. Using Stored Procedures

For certain types of databases, stored procedures may be used as a source of chart data in the same way as tables and views. Stored procedures offer several advantages over tables and views, the primary one being that more complex logic can be used in building a dataset than is possible with a simple SELECT statement. Not every stored procedure that exists in a database, however, can be used as a source for chart data. For example, a procedure must return a result set containing at least two columns for it to be suitable as a source of chart data. Also, the use of stored procedures that are long-running (more than 5 seconds), or have significant side effects (i.e. they update one or more tables in the database), should be avoided.

In order to use a particular stored procedure as a source of chart data, it must first be configured in the iDashboards repository. Once configured it will appear in the list of tables and views for its parent Data Source in the iDashboards Build application. Stored procedures that appear in the list will be grouped together below the tables and views.

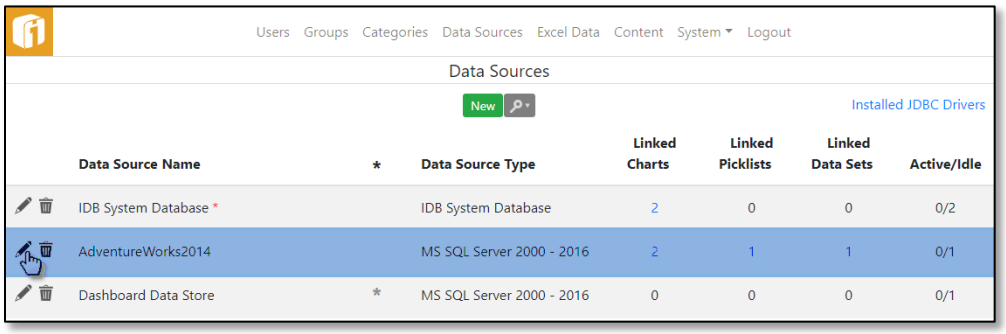
> *Note: Currently, stored procedures may only be used with Microsoft SQL Server, MySQL and Oracle databases.*

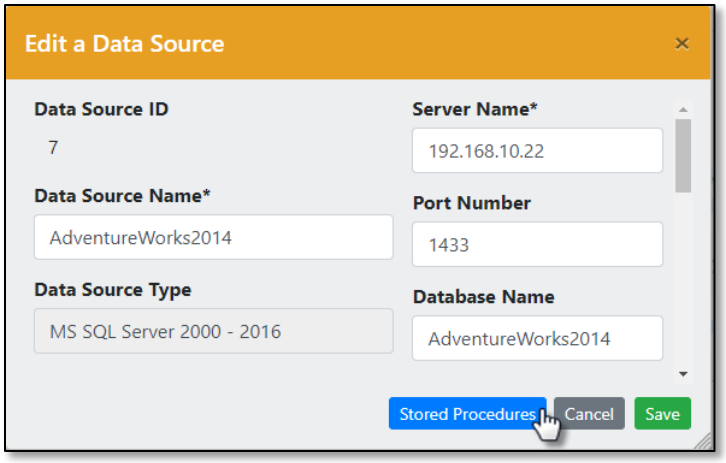
12.1.1 Configuring a Stored Procedure

The first step in configuring a stored procedure is to access the stored procedures screen for the database containing the procedure.

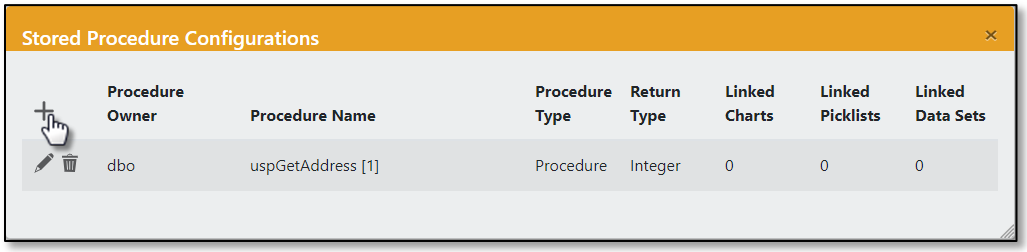
From the Data Sources screen, chose the database where the stored procedure is located.

In this case it is the "AdventureWorks2014". To edit a data source, select its Edit icon ( ). The select the "Stored Procedures" button.

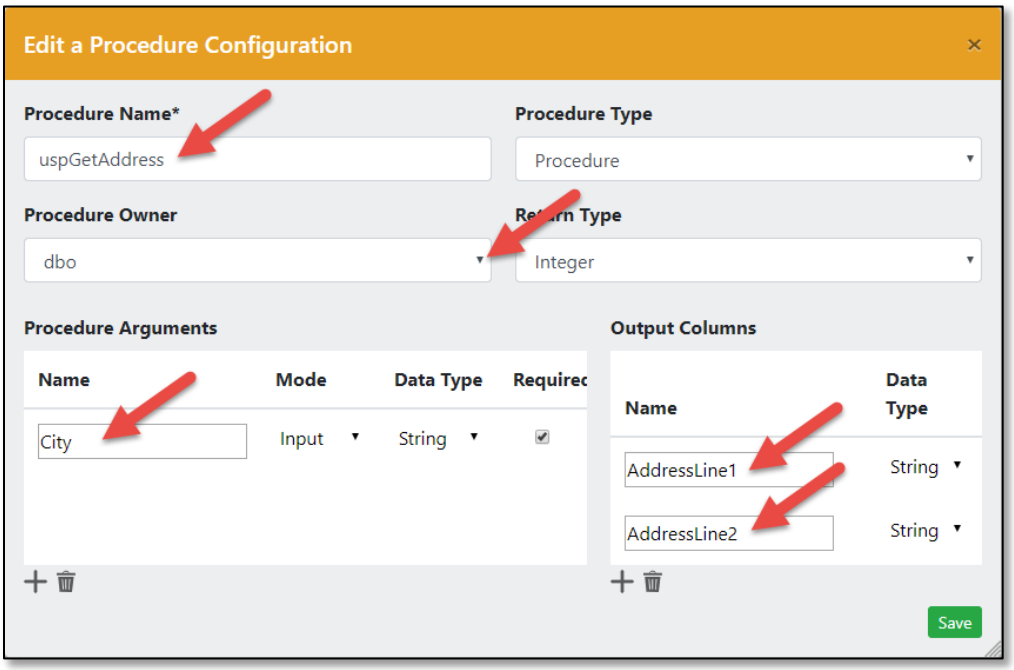




The "Stored Procedures" button will open the "Stored Procedure Configurations" dialog.

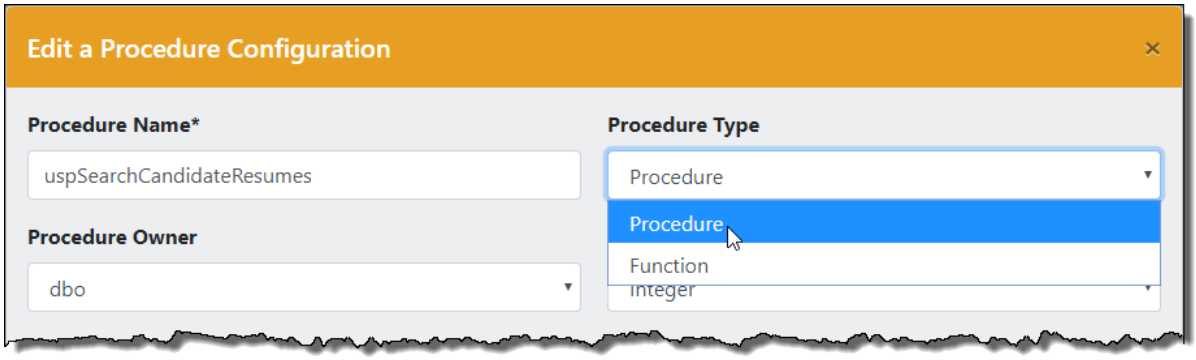


To create a new store procedure configuration, select the "Add" icon (+). This will open the "Edit a Procedure Configuration" dialog.



The dialog contains the following inputs:

- **Procedure Name** - The name of the stored procedure should be entered in this field, using the correct case. For Oracle databases, the name of the function should be qualified by the package name, for example *package_name.function_name*.
- **Procedure Type** - For some databases, multiple "types" of stored procedures may be used as sources of chart data. For example, with Microsoft SQL Server, both normal procedures and inline table-valued functions may be used. For such databases, there will be a dropdown from which the appropriate procedure type must be selected before it can be configured. For databases for which only a single procedure type is supported (Oracle, for example, for which only stored functions are supported) the dropdown will not be present.



- **Procedure Owner** - This is the name of the database user or schema that owns the stored procedure. The Owner dropdown will only list those users/schemas that: a.) Conform to the schema name pattern used (if any) when the Data Source was created, and b.) Contain at least one table or view which is visible to the user account used to connect to the database.
- **Return Type** - If multiple return types are possible for the selected procedure type, they will be listed in a dropdown in the Return Type field, and the correct one must be selected. If only one return type is possible, the dropdown will be omitted and that return type will appear as static text.
- **Procedure Arguments** - If a procedure takes arguments, they should be entered in the "Procedure Arguments" section, in the correct order. Select the "Add" icon (+) to create a new argument line. Users building charts will have the ability to enter values for each argument. Descriptions of the inputs within the Procedure Arguments section follow.
    - **Name** - This is the name used to identify an argument. It does not have to be the actual name of the argument, but rather, it can be a descriptive name that will be helpful to users building charts.
    - **Mode** - This indicates whether a procedure argument is strictly an input argument, or whether it is used for both input and output. Any output values are ignored when a procedure is executed. Pure output arguments are not supported.

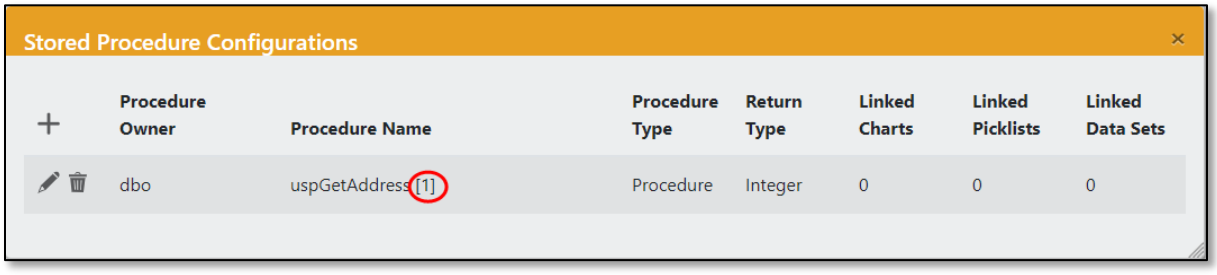
- **Data Type** - The "data type" of an argument should be the generic type — Integer, Number or String — that best corresponds to the argument's actual data type. Date or datetime arguments are not supported; instead, procedures should accept dates as String-type arguments (such as VARCHAR).
- **Required** - This box should be checked if NULL cannot or should not be passed for a procedure argument. It can be left unchecked if NULL values are acceptable.

- **Output Columns** - This section of the dialog is where the output columns of the stored procedure are entered. Select the "Add" icon (+) to create a new output column line. Each output column of the procedure must be entered, in the correct order. Descriptions of the inputs within this section follow.
  - **Name -** This is the name of a column to be added to the list of output columns. The name given to each output column does not have to be the actual name, but rather, it can be a descriptive name that will be helpful to users building charts.
  - **Data Type -** The "data type" of an output column should be the generic type — Integer, Number or String — that best corresponds to the column's actual data type. (String should be used for date or datetime columns.)

The list of output columns or procedure arguments can be reordered by selecting the row and dragging it to a new location. They can individually be deleted by selecting its "Delete" icon (🗑).

After all of the required values on the procedure edit dialog have been provided, selecting the "Save" button will add the procedure configuration to the repository database and make it available for building charts. The procedure edit dialog will remain displayed, allowing further modifications to the procedure's configuration, until the "Save Changes", "Update Procedure", or "Discard Changes" button has been selected. (The button label changes depending on the state of the screen.)

When the "Edit a Procedure Configuration" dialog is dismissed, the newly added procedure will appear in the list of procedures configured for that database.



The procedure name is followed by a number in square brackets. This is the ID number that iDashboards uses internally to identify the procedure configuration. It is displayed along with the procedure name, both in the Administrator application and the iDashboards Build application, to distinguish it from other stored procedures having the same name but a different argument list. (Some databases, such as Oracle, allow a function or procedure name to be "overloaded" in this manner.)

Now, from the iDashboards Build application, you may create a new chart using the "Stored Procedure" Data Set type.

12.1.2 Modifying a Stored Procedure Configuration

> *Note: Modifying a stored procedure which has already been used as a source of chart data can cause dependent charts to malfunction, and should be done cautiously.*

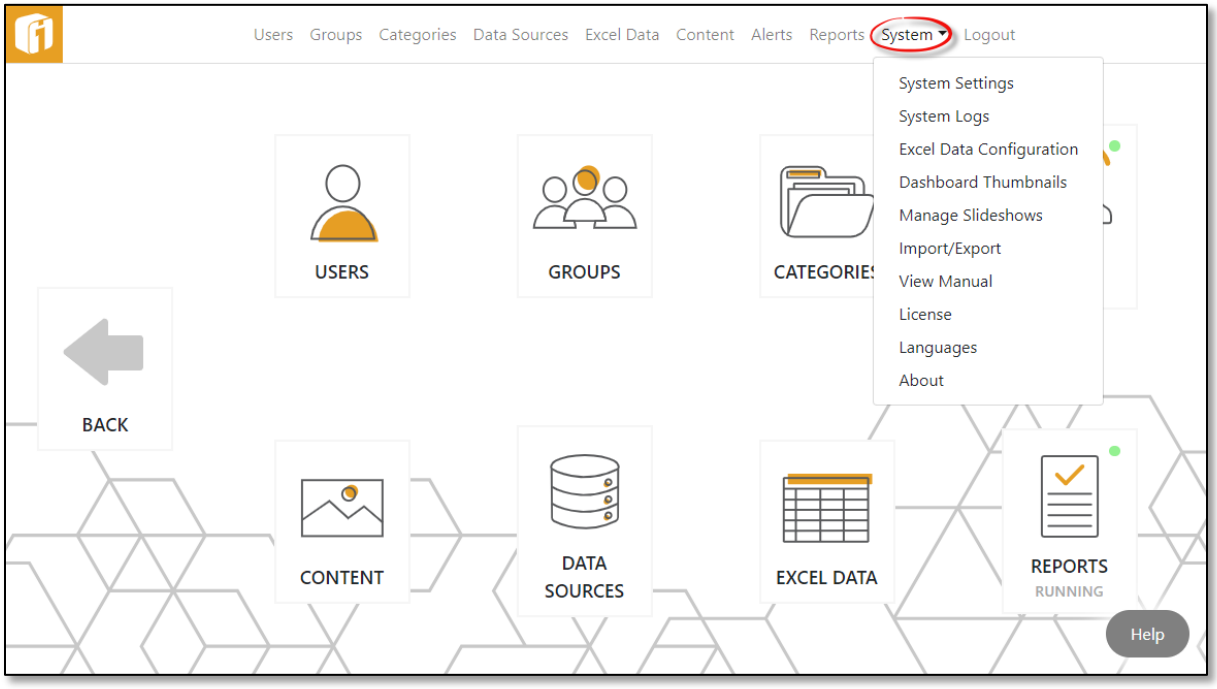
To modify an existing stored procedure configuration, select its Edit icon () on the "Stored Procedure Configurations" screen for its parent Data Source. The "Edit a Procedure Configuration" screen will appear, through which the procedure's properties can be modified. Select the "Save" button to save the changes. Select the "Cancel" icon (✕), in the top right corner, to discard any changes and return to the "Stored Procedure Configurations".

12.1.3 Removing a Stored Procedure Configuration

A stored procedure configuration can be removed from the repository, as long as there are no charts or picklists that depend on it. To remove a stored procedure configuration, select its "Delete" icon (), and then select "OK" on the confirmation dialog that appears.

13. System Configuration

Various aspects of the iDashboards server can be controlled through the system configuration screens of the Administrator application. These screens are accessed by selecting "System" on the menu bar.



System configuration is divided broadly into ten areas in a drop-down list. System Settings, System Logs settings, Imported Data Configuration, Dashboard Thumbnail configuration, Manage Slideshows, Import/Export, View Manual, License file management and Language setup.

Available settings may differ from one version or installation of iDashboards to another, and may not match the list of system properties documented here.

13.1 Modifying a System Setting

To modify a system setting, select its Edit icon ( ). A dialog will appear through which the setting can be edited. For most system settings, the value must be entered into a textbox, while for some, the value can be selected from a dropdown list. The edit form for a system setting will include a description of that setting and its valid values.

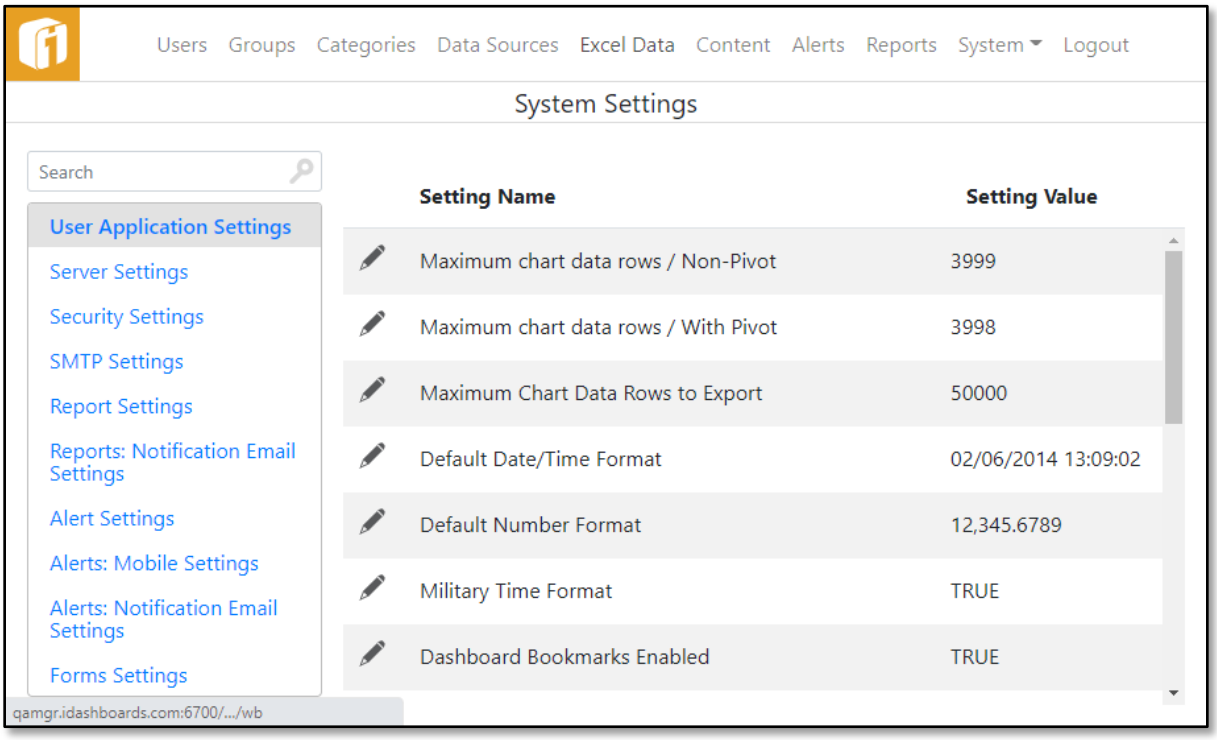
After a system setting has been modified, select the "Save" button to save the changes, or "Cancel" icon (✕) to discard the changes and return to the system settings screen.

13.2 System Settings

System Settings are global settings used to control the iDashboards server's behavior and user experience. They are managed through the System Settings screen. The System Settings screen can be accessed by selecting the "System Settings" from the "System" drop-down list from any other screen in the Administrator application.

System settings are grouped according to their function into setting categories. A list of the available Setting Categories appears on the left side of the System Settings screen. When a category is selected, a list of the system settings for that category appears on the System Settings screen.

In a typical installation, each system setting can be left at its default value. In some instances, however, a system setting might need to be changed to accommodate aspects of the user experience, the host application server, or the repository database.



13.2.1 User Application Settings

These are settings related to user interface functionality.

- **Maximum chart data rows / Non-Pivot:** This is the maximum number of data rows that will be returned for a chart when no pivot column has been selected. Allowable values are from 100 to 1000. The default value is 1000. Decreasing this value may improve performance, but restricts the maximum rows a chart can display.
- **Maximum chart data rows / With Pivot:** This is the maximum number of data rows that will be returned for a chart that has one or more pivot columns. Allowable values

are from 300 to 3000. The default value is 3000. Decreasing this value may improve performance, but restricts the maximum rows a chart can display.

- **Maximum Chart Data Rows to Export:** This is the maximum number of data rows that will be exported. The default value is 50,000.
- **Default Date/Time Format:** This setting indicates the default format in which date or date/time values will be displayed in charts when no chart-specific format has been specified. The default value is M/d/YY.
    - The values shown in the dropdown list are samples of how the date/time value of Sunday, February 6, 2005 1:09:02 PM would be displayed on a chart.
- **Default Number Format:** This setting indicates the default number format that will be assigned to newly created charts. Through the User Interface, the number formatting can be changed for each chart. The number format is saved with a chart, so changing this setting will have no effect on existing charts. The default value "12,345.6789" is equivalent to a "floating decimal", which automatically suppresses decimal places if the data set returns only whole numbers.
    - "Comma" Thousands Separator then "Period" Decimal Separator

    Or

    - "Period" Thousands Separator then "Comma" Decimal Separator
- **Military Time Format:** This setting indicates whether or not time selection will be displayed in the military format with hours from 0-23 rather than with an AM/PM designation. Changing this setting will affect the "Time" component of input parameters. The default value is TRUE.
- **Dashboard Bookmarks Enabled:** This setting indicates if the user may set and access dashboard bookmarks. The default value is TRUE.
- **Prevent Session Timeout:** When this setting is TRUE, the server session will not timeout due to client communication inactivity. A value of FALSE indicates the session will time out after the number of minutes indicated in the Session Timeout setting. The default value is FALSE.
- **Session Timeout (minutes):** Session Timeout value in minutes.
- **Data Import - Maximum File Size:** This is the maximum allowed size (in bytes) for Excel or CSV files that are imported as chart data. Allowable values are from 1000 to 9,999,999. (Do not enter commas.) The default value is 9999999.

- **Case-Sensitive Filter Criteria:** This setting determines whether or not the filter criteria applied to String columns of stored procedure and data feed results, and during alert checks and drilldowns, will be case-sensitive. The default value is TRUE.
- **CSV Export – Decimal Separator**: This setting indicates the character that is used as the decimal separator in numbers when chart data is exported to a CSV file. When Decimal Point is selected, commas are used as delimiters between data values. When Decimal Comma is selected, semicolons are used as delimiters between data values. Individual users can override this setting according to their preferences. The default value is "Decimal Point (.)".
- **Viewer Role May Export:** If true, users with the Viewer role will be able to export chart data in CSV format. The default value is TRUE.
- **Guest User May Export:** If true, guest users will be able to export chart data in CSV format. The default value is FALSE.

  Note: This applies when a guest user has done a guest login. If they login using a username/password, their permissions will default to the user role they are assigned.
- **Viewer Role May View Diagnostics:** If true, users with the Viewer role will be able to view the chart data diagnostics. The default value is FALSE.
- **Guest User May View Diagnostics:** If true, guest users will be able to view the chart data diagnostics. The default value is FALSE.
- **Viewer Mode May Export:** If true, then when the iDashboards application is invoked in viewer mode chart data can be exported in CSV format.

13.2.2 Server Settings

These are settings related to server functionality.

- **Generate Global IDs:** If true, each dashboard, chart, category and data source will be assigned a global ID upon creation. The default and recommended value is FALSE.
- **Custom Query Timeout (Seconds):** This setting indicates the maximum time, in seconds, that a custom query used to produce chart data will be allowed to run before it is terminated. Allowable values are from 1 to 120. The default value is 10.
- **Database Maximum Transaction Isolation Level:** This setting should be left at "Serializable" unless you encounter error messages that say "Error setting isolation level" while doing things that modify the iDashboards repository database, such as adding or changing charts, dashboards, users, etc. The default value is Database Default.

If this type of error occurs, change this setting to reflect the maximum isolation level supported by your iDashboards repository database. (The levels increase toward the bottom of the list.) If you're not sure what the maximum supported isolation level is, use the "Database Default" setting.

- **Clustered Appserver:** Set this property to YES if iDashboards is running in a clustered application server, and NO if it is not. The default value is NO.
- **Clustered Cache Max Age (Seconds):** If iDashboards is running in a clustered application server, this is the maximum time, in seconds, that system settings will be cached before being refreshed from the database. The default value is 300.
- **Data Import - Temporary Directory:** When a user uploads an Excel or CSV file as imported chart data, iDashboards might need to temporarily write it to disk if it unusually large. By default, it will write it to the system temporary directory, but this setting can be used to indicate a different directory. The entered path can be absolute or relative to the appserver's working directory, as long as it exists and is writeable. The directory value is blank by default.

  If iDashboards is running in a clustered appserver, the entered path must be valid for *all* nodes in the cluster. Therefore, it's recommended to leave this setting blank in a clustered environment.
- **Proxy Server Enabled**: If true, a proxy server can be used to load external images residing on the Web into a dashboard. During the dashboard design, the window to browse for Images or other Content will have an option titled "Use Proxy Server". The user must check this box to use a URL outside of the iDashboards network. The default value is "FALSE".
- **Request Logging Enabled**: If true, request logging can be used to create a log of the commands sent to the server. All user activities within the admin and user console such as (login, chart load etc.) will be logged to a built-in table within the repository namely "fv_request_log" table. The default value is "FALSE".
- **Request Logging Retention (Days)**: If Request Logging Enabled is set, this settings is used to define the maximum time, in days, that the server request logs will be stored before being deleted from the database. The default value is "FALSE".
- **Log Remote Host Name**: If true, the remote hostname of the requesting host will be stored in the request log, otherwise only the IP address will be stored. The default value is "TRUE".
- **Imported File Auto Upload Enabled**: This setting determines whether or not users can use the Auto Uploader desktop application to upload workbooks and text files to this server application. The default value is "TRUE".

- **Imported File Auto Upload – Minimum Required Role**: This is the minimum role a user account must have in order to use the Auto Uploader desktop application. The default value is "Builder".

13.2.3 Security Settings

These are settings related to server functionality.

- **Override Security on Dashboard Drilldown:** If this setting is true, users may access dashboards through drilldowns that they would not otherwise have access to. This makes the behavior of dashboard drilldowns similar to that of chart drilldowns, in which users may drill down to any chart regardless of their groups' access to its category. The default value is FALSE.

- **Data Source Access Control Enabled:** This setting determines whether or not access to data sources is controlled at the user group level. If the setting is TRUE, then additional screens will be available in the DATA SOURCES and GROUPS sections of the Administrator application, through which groups can be granted (or denied) access to individual data sources for the purpose of building charts. Note that groups will still be able to view or modify a chart based on their access level to the chart's category, even when the chart is based upon a data source to which they have been denied access. The default value is FALSE.

- **Allow Password Change:** This setting determines whether or not users can change their iDashboards passwords through the user settings dialog of the application. The default value is TRUE.

  Note: If an authentication module other than the default iDashboards authentication module is being used, (for instance, if a user's login credentials are being authenticated against an Active Directory domain) then password changes will be disabled regardless of the value of this setting.

- **Require User Email:** This property indicates whether or not users are allowed to be created without an email. The default value is FALSE.

- **Allow Null User Passwords:** This setting determines whether or not a user can be added through the user admin screen with a null (empty) password. This should only be set to TRUE when an external authentication module is in use, and any passwords stored with a user record in the iDashboards repository are ignored. If an external authentication module is not in use, setting a user's password to null will effectively disable logins for that account. Regardless of the authentication mechanism being used, a user is never able to log into iDashboards by presenting a null password. The default value is FALSE.

- **Allow Auto-Completion of Passwords:** This setting indicates whether or not iDashboards passwords can be stored by the browser and automatically supplied by the browser upon logging into the iDashboards application. The default value is TRUE.

- **Filter-On-User Style:** This property determines the behavior of charts that have a "Filter on User" column set, when no matching filters exist on the FV_USER_FILTER table for that user. The default value is "Strict".

- **User Application X-Frame-Options Header:** This property indicates to the browser whether or not the iDashboards application can be framed by another web page. Setting this property ensures the iDashboards application cannot be embedded into other web pages.

  If the selected value of this property is "None", then the X-Frame-Options response header will not be set, allowing the iDashboards application to be framed by another page. If the selected value is "Deny", then the iDashboards application can never be framed by another web page. If the selected value is "Same Origin", then the iDashboards application can only be framed by a web page from the same domain. The default value is "None".

- **Valid iframe Source Hosts:** This setting consists of one or more hostnames (one per line) that indicate the valid hosts that can be used with the **iframe:** macro when drilling down to a web page. Some websites, such as the Google Maps Embed API, require their content to be displayed within an iframe. In the example of using the Google Maps Embed API, the administrator could add "www.google.com" as a valid source host.

  This setting primarily applies toward URL requests from the iDashboards View interface. To use the "iframe:" macro, prefix the URL in the drilldown tab, within the Chart Designer, with "iframe:". This will embed the URL (making it the *src* of an iframe HTML element) within the chart.

- **Destroy Session on Login - User Application:** This property indicates whether or not an existing browser session should be destroyed when logging into the iDashboards User Application.

- **Destroy Session on SSO Login - User Application:** This property indicates whether or not an existing browser session should be destroyed when logging into the iDashboards User Application.

- **Destroy Session on Login - Admin Application:** This property indicates whether or not an existing browser session should be destroyed when logging into the iDashboards Admin Application.

- **IFRAME Dashboard Panels Enabled:** This determines whether or not IFRAME dashboard panels will be displayed in dashboards. These dashboard panels display web content from external servers in an IFRAME element, and they could potentially be used for malicious purposes.

- **Require Secure Connection For Sharing:** If this setting is "Yes", then the connection associated with the requests for shared data sets must be secure. If this

setting is "No", then the connections associated with the request is not required to be secure.

- **Override Security for Comment Moderation:** If this setting is TRUE, moderators can view articles and dashboards with flagged comments regardless of permissions.

13.2.4 SMTP Settings

iDashboards Alerts and Reports Servers, as well as Password Reset, uses an external SMTP service to send emails. On the SMTP Settings screen, locate the following settings:

- **SMTP Host:** This is the hostname or IP address of the machine on which the SMTP service is running. If this value is not provided 'Forgot Password?' will not appear on login screen.
- **SMTP Port:** This is the number of the TCP/IP port on which the SMTP service is listening. (The standard SMTP port number is 25.)
- **SMTP Service Requires Authentication:** Set this to "Yes" if the SMTP service requires authentication or incoming connections, or to "No" if it does not.
- **SMTP Service User:** If the SMTP service requires authentication, this setting must contain the username of the user that will be used to connect to it, otherwise it should be left blank.
- **SMTP Service Password:** If the SMTP service requires authentication, this setting must contain the password that will be used to connect to it, otherwise it should be left blank.
- **SMTP Encryption:** This setting determines the type of encryption (if any) used to secure the connection with the remote mail server. The options are "None", "SSL (Secure Socket Layer)" and "TLS (Transport Layer Security)".

13.2.5 Report Settings

- **Reports Enabled:** This setting determines whether reports can be viewed and scheduled in the iDashboards user interface. Values are True (default) and False.
- **Report Scheduling Enabled:** This setting determines whether reports can be scheduled in the iDashboards user interface. Values are True (default) and False.
- **Server Startup State:** This setting determines the initial state of the server upon startup. The two possible values are:
    - Running (default): The Report Schedule Thread will be started, and the server will check for reports according to its schedule.
    - Paused: The Reports Schedule Thread will be in the paused state when the server starts up. It will need to be started manually through the Server Status screen of the Reports Admin application.
- **Reports Enabled By Default:** This setting determines whether or not reports are enabled by default for newly created charts or dashboards. Values are True (default) and False.
- **Viewer Role May Run Reports:** If true, users with the Viewer role will be able to run reports. Values are True (default) and False.

- **Guest User May Run Reports:** If true, guest users will be able to run reports. Values are True and False (default).

13.2.6 Reports: Notification Email Settings

- **Notification Email Enabled:** If this setting is "No", then all email notifications, for reports and server event notifications, will be disabled. If it is "Yes", then the settings identified in the 13.2.4 "SMTP Settings" must be properly configured to connect to the SMTP Service. Default is "No".
- **Notification Email "From" Address:** This setting must be a valid email address that will appear in the "From" header of notification email messages, for example "reports@mycompany.com". This setting is required if email notifications are enabled.
- **Notification Email "From" Name:** This optional setting is the name that will appear before the email address in the "From" header of notification email messages, for example "iDashboards Report".
- **Alert Notification Subject:** This optional setting is a string that will be used to build the subject line for alert notification emails.
- **Server Events Notification Threshold:** This setting determines what types of server events will generate emails to the addresses listed in the Server Events Notification List setting. The higher in the list a selection is, the fewer notification emails will be sent. Since a selection represents a threshold, each selection implicitly includes the ones above it.

  Selecting "Disabled" will turn off all email notification of server events.

  INFO-level events include the server starting up or being restarted after a pause. WARNING-level events include the server being shut down (in an orderly manner by the application server) or paused. ERROR-level events include any conditions that prevent the iDashboards Reports Server from functioning properly.

- **Server Events Notification List:** This setting is a list of email addresses that will receive notifications of server events. Each email address should be on a separate line. Email addresses may be in plain format, for example:

  jsmith@company.com

  or in any RFC822-compliant format, for example:

  "Jane C. Smith" jsmith@company.com

  The combined length for all email addresses, including end-of-line characters, must not exceed 500 characters.

- **Server Event Subject:** This optional setting is a string that will be used to build the subject line in server event notification emails.

- **Server Error Subject:** This optional setting is a string that will be used to build the subject line in server error notification emails.

13.2.7 Alert Settings

- **Server Startup State:** This setting determines the initial state of the server upon startup. The two possible values are:
    - Running (default) – The Alert Monitor Thread will be started, and the server will check for alerts according to its schedule.
    - Paused – The Alert Monitor Thread will be in the paused state when the server starts up. It will need to be started manually through the STATUS screen of the Alerts Admin application.
- **Alert Instance Retention (Days):** This setting indicates the number of days an alert instance will be kept before it "ages out" of the alert queue and is deleted from the repository database. Allowable values are from 1 to 9999. If the setting is left blank, then alert instances will remain in the queue indefinitely. This setting will not remove or alter alert configurations. The default is 90.
- **Browser Alert Check Interval (Minutes):** This setting indicates the interval, in minutes, in which a user's Alerts dashboard will check for new alert instances. Allowable values are from 1 to 60. The default is 1 minute.
- **Maximum Displayed Alert Instances:** This setting indicates the maximum number of alert instances that will be displayed in a user's Alerts dashboard. If the number of alert instances for a user exceeds this maximum, the newest ones will be given priority. Allowable values are from 20 to 200.The default is 50 instances.

13.2.8 Alerts: Mobile Settings

A mobile SMS text message can also be used to notify users when an alert has triggered.

- **Mobile Notifications Enabled:** If this setting is "No", then all mobile SMS text notifications will be disabled. If it is "Yes", then mobile SMS text notifications will be sent for alerts that have send a text message enabled, to users that have a mobile phone number configured and elected to receive SMS notifications.
- **Maximum Number of Segments:** This is the maximum number of segments of the complete SMS text message that will be sent to the user's phone. Allowable values are from 1 to 99. The default value is 10. Larger values may result in lot of SMS notifications being sent out to accommodate the entire message.
- **Mobile Segment Prefix Enabled:** If this setting is "Yes", then each segment of a long SMS text message will be prefixed with the index number and total number of segments (e.g. (1/10) indicates first segment out of 10 segments of a long SMS text message). If it is "No", then SMS segments will not be prefixed with indices.

13.2.9 Alerts: Notification Email Settings

- **Notification Email Enabled:** If this setting is "No", then all email notifications, for alerts and server event notifications, will be disabled. If it is "Yes", then the settings

identified in the 13.2.4 "SMTP Settings" must be properly configured to connect to the SMTP Service.

- **Notification Email "From" Address:** This setting must be a valid email address that will appear in the "From" header of notification email messages, for example "alerts@mycompany.com". This setting is required if email notifications are enabled.
- **Notification Email "From" Name:** This optional setting is the name that will appear before the email address in the "From" header of notification email messages, for example "iDashboards Alerts".
- **Alert Notification Subject:** This optional setting is a string that will be used to build the subject line for alert notification emails.
- **Server Events Notification Threshold:** This setting determines what types of server events will generate emails to the addresses listed in the Server Events Notification List setting. The higher in the list a selection is, the fewer notification emails will be sent. Since a selection represents a threshold, each selection implicitly includes the ones above it.

  Selecting "Disabled" will turn off all email notification of server events.

  INFO-level events include the server starting up or being restarted after a pause. WARNING-level events include the server being shut down (in an orderly manner by the application server) or paused. ERROR-level events include any conditions that prevent the iDashboards Alerts Server from functioning properly.

- **Server Events Notification List:** This setting is a list of email addresses that will receive notifications of server events. Each email address should be on a separate line. Email addresses may be in plain format, for example:

  jsmith@company.com

  or in any RFC822-compliant format, for example:

  "Jane C. Smith" jsmith@company.com

  The combined length for all email addresses, including end-of-line characters, must not exceed 500 characters.

- **Server Event Subject:** This optional setting is a string that will be used to build the subject line in server event notification emails.
- **Server Error Subject:** This optional setting is a string that will be used to build the subject line in server error notification emails.

13.2.10 Forms Settings

- Forms Enabled**:** This setting determines whether data forms will appear in the iDashboards user interface.

13.2.11 Knowledge Base Settings

Note: The Knowledge Base feature must be enabled within the iDashboards license. The administrator has the option to disable the entire Knowledge Base feature.

- **Knowledge Base Enable:** This setting determines whether the Knowledge Base will appear in the iDashboards user interface.
- **Require Authentication to View Articles:** This setting determines whether authentication is required to view articles.
- **Hide Change Information:** This setting determines whether to show the created and updated user information.
- **Comment Sorting:** This setting determines the order that the comments are shown and where the new comment box is located.
- **Show Comments on Public Articles:** This setting determines if article comments will be displayed for publicly-accessible articles.

13.2.12 Public Access Settings

- **Public Access URLs Only:** This setting determines if only the configured public access URLs will be available or if public access can be provided by other means, such as a legacy URL.
- **Show Comments on Public Dashboards:** This setting determines if dashboard comments will be displayed for publicly-accessible dashboards.

13.3 Authentication Settings

Built into the core of iDashboards, is a framework to handle user authentication. Additionally, a system administrator can configure alternative authentication methods to assist with system integration or login automation needs.

Some methods described here require file access to the application server, some methods have a user-interface, and some require a combination of both.

13.3.1 External Authentication

When logging into iDashboards, a user must provide a username and password. If the username is that of a valid iDashboards user, and the password is the correct one for that username, the user will be granted access to iDashboards. This process is referred to as *authentication*.

In its default configuration, iDashboards authenticates a username/password pair (referred to as *login credentials*) by searching for a user record with a matching pair in the repository database. One of the drawbacks of this arrangement is that iDashboards users must maintain a separate password for their iDashboards account, in addition to the ones they use to log into a company network or other applications. Also, when an iDashboards user leaves an organization, an administrator must disable or delete the user's iDashboards account in addition to other login accounts.

The iDashboards server can be configured so that it delegates the authentication of a user's login credentials to some external system. There are several possible advantages to using an external authentication mechanism, including:

- Users do not need to maintain a separate iDashboards password.
- If the external system enforces password expiration and/or change policies, they will apply to iDashboards also.
- Security measures provided by the external system, such as account lockout after a specific number of unsuccessful login attempts, will be applied to iDashboards users as well.
- Certain types of user account administration, such as disabling a user's account or resetting the password, would only need to be done in the external system.

External user authentication is accomplished through the use of *authentication modules*. An authentication module consists of one or more Java classes that implement a small part of the Java Authentication and Authorization Service (JAAS) API (http://support.idashboards.com/links/jaas). There are three possible sources for authentication modules that can be used by iDashboards:

- An authentication module that is bundled with iDashboards can be used. Currently, only a single authentication module is included with iDashboards. This authentication module, referred to as the LDAP authentication module, is designed to authenticate a user's login credentials against a directory server, such as Microsoft Active

Directory or Novell eDirectory, using LDAP (Lightweight Directory Access Protocol) to communicate.
- A third party authentication module can be used. There are a number of commercial or open-source JAAS-compliant authentication modules available.
- A custom authentication module can be developed. This may be the only viable option when the system used for external authentication cannot be accessed through standard APIs or protocols.

13.3.1.1 Limitations of External Authentication Modules

Before configuring an iDashboards server to use an external authentication module, the following limitations should be understood:

- A user record must still be created for each iDashboards user. If the user record does not exist for a given username, a login will fail even if the login credentials are validated by the external authentication system.

External authentication modules are used for authentication only. They do not affect the access levels a user has within the iDashboards system. Access levels are determined by a user's role and group, which are configured through the user maintenance screens.

- Although the JAAS API provides for "stacking" multiple authentication modules, so that a successful authentication by any module will result in a successful login, this feature is not supported by iDashboards. Only a single authentication module may be configured, and it must successfully authenticate a user in order for a login to succeed.
- The iDashboards system user account, which has the username "admin", will always be authenticated through the normal iDashboards authentication process, regardless of any external authentication module that may be in use. This insures that the admin user can always log into iDashboards even if the external system is unavailable.

13.3.1.2 Configuring an External Authentication Module

> *Note: External authentication is an advanced feature of iDashboards, and configuring iDashboards for external authentication requires knowledge of certain aspects of Java-based systems and the JAAS API. This document was written with the assumption that the reader has the requisite knowledge, or will acquire it from other sources.*

As mentioned previously, an iDashboards authentication module consists of one or more Java classes that implement a small part of the JAAS API. The specific part that must be implemented is the javax.security.auth.spi.LoginModule interface (http://support.idashboards.com/links/loginmoduleinterface), referred to simply as the LoginModule interface. The class that implements the LoginModule interface, (referred to as the LoginModule class) along with any classes on which it depends, must be in the iDashboards application server's classpath, or they can be packaged in a JAR file and placed in the <IVIZGROUP HOME>\drivers directory.

A LoginModule class may optionally require one or more "options", which are name-value pairs that are passed to it in a Java Map object as the "options" argument of its *initialize* method. The names and possible values of these options are specific to each LoginModule class, and should be described in the documentation for the LoginModule class.

Once the necessary classes have been added to the classpath, the following steps, which can be performed with the iDashboards server running, will configure iDashboards to use the authentication module:

1. Add an entry named "login.module" to the ivizgroup.properties file that provides the fully-qualified class name of the LoginModule class. For example, if the LDAP authentication module bundled with iDashboards is being used, the login.module entry would look like this:

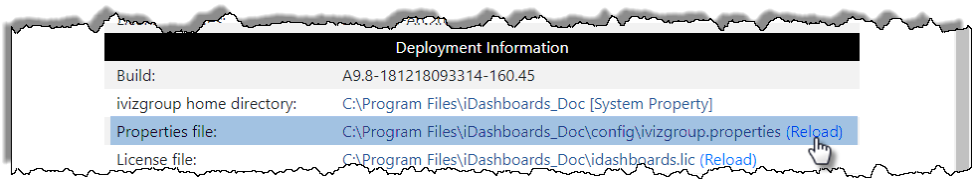
   ```
   login.module=com.ivizgroup.idb.security.auth.LDAPLoginModule
   ```

2. Add an entry for each module-specific option to the ivizgroup.properties file, with each option name prefixed with "login.module". For example, if the LoginModule class takes an option called "debug" with possible values "true", or "false", the entry in ivizgroup.properties would look like:

   ```
   login.module.debug=true
   ```

   When the iDashboards module creates an instance of the configured LoginModule class, it will read all of the properties from ivizgroup.properties whose names begin with "login.module.", strip the "login.module." portion from the beginning of each name, put the name value pairs into a Java Map object, and pass the Map to the instance's *initialize* method as the "options" argument.

3. Reload the ivizgroup.properties file by selecting the "Reload" link that appears on the menu item "System > About" screen.



Once the ivizgroup.properties file has been reloaded with the new settings, the iDashboards server will use the newly-configured login modules for all user logins, with the exception of the "admin" account. If problems are encountered, the idashboards.log file should be checked for error messages that can be used to troubleshoot the problem. If changes to the ivizgroup.properties file are required, they will take effect when the "reload" link is selected.

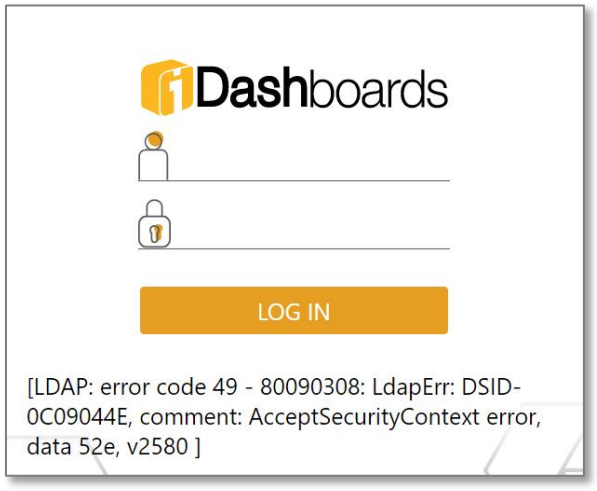
13.3.1.3 Login Message Mapping

When a login attempt against an external authentication module is unsuccessful, the error message returned by the authentication module may not be suitable for display to users. For example, if the bundled LDAP authentication module is configured to authenticate users against an Oracle Internet Directory (OID) server, and a user enters an incorrect username or password, the following error message will be returned by OID:

```
[LDAP: error code 49 - Invalid Credentials]
```

Under the default configuration, this message will be displayed to the user.

In cases where such error messages would not adequately convey the cause of the login failure to the user, a "message mapping" can be configured in an XML file, referred to as the login message mapping file. If the file exists and is properly formatted, the iDashboards server will examine the contents of the authentication module's error message (referred to here as the "module message"), and look in the mapping file for a more user-friendly message that might be "mapped" to the particular module message. If one is found, it will be displayed to the user, and if not, the module message will be displayed.



13.3.1.4 Name and Location of the Message Mapping File

By default, the name of the message mapping file is "loginmessages.xml", and iDashboards will look for it in the <IVIZGROUP HOME>\config directory. An alternate name and location can be specified by the system setting "Login Message Mapping File", which is in the "External Authentication" setting category.

13.3.1.5 Caching the Message Mapping File

To improve performance, the iDashboards server will cache the contents of the message mapping file in memory. To make changes to the file take effect without restarting the application server, the caching must be turned off. This is controlled by a system setting, "Cache Login Message Mapping File", which is in the "External Authentication" setting category.

13.3.1.6 Creating the Message Mapping File

The root element of the message mappings file is named ***message-mappings***. The message-mappings element has a single optional attribute named case-sensitive, which has the possible values true or false. The case-sensitive attribute provides a global default value indicating whether or not the matching of module messages will be case-sensitive. If the case-sensitive attribute is omitted, then the default will be true; in other words, matching will be case-sensitive by default. Shown below is the shell of a message mapping file, with the case-sensitive attribute set to false.

```
<?xml version="1.0" encoding="UTF-8"?>

<message-mappings case-sensitive="false">

</message-mappings>
```

The message-mappings element must contain one or more mapping elements. A mapping element defines a mapping between one or more module messages and a single user message. ("User message" refers to the message that will be displayed to the login user.) The mapping element contains two mandatory child elements, the match element and the message element.

The match element contains a string, called the "match string", for which the iDashboards server will search a module message in order to identify the module message. The match element may have two optional attributes:

1. Location - The possible values for the location attribute are begin, end, any or all, and if omitted, the default value is any. This attribute determines how the module message will be searched for the match string:
   - **begin** — Indicates that the match string must appear at the beginning of a module message for a match to occur.
   - **end** — Indicates that the match string must appear at the end of a module message for a match to occur.
   - **any** — Indicates that the match string may occur anywhere in the module message for a match to occur.
   - **all** — Indicates that the match string must match the entire module message in order for a match to occur.
2. case-sensitive - The possible values for this attribute are true or false. If present, the value of this attribute will override the global default case-sensitivity for the individual mapping. If omitted, the global default case-sensitivity will be applied when checking for matches.

Building upon the example above, shown below is a message mapping file that contains a single mapping element:

```
<?xml version="1.0" encoding="UTF-8"?>

<message-mappings case-sensitive="false">
```

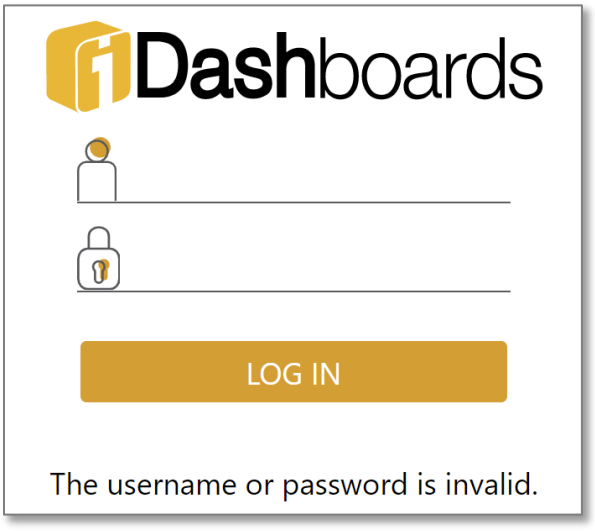
```
    <mapping>
    <match location="end" case-sensitive="true">
   error code 49 - Invalid Credentials]
    </match>
    <message>The username or password is invalid.</message>
    </mapping>
</message-mappings>
```

In the example above, the default value of false for case-sensitivity is overridden by the case-sensitive attribute on the match element. Note also that leading and trailing whitespace is trimmed from the contents of both the match and message elements before matching are performed.

13.3.1.7 How it Works

When a user login fails, the iDashboards server evaluates the module message against each mapping defined in the message mapping file, in the order in which they appear. When a match is found, i.e. the match string is found in the module message within the constraints defined by the location and case-sensitive attributes of the match elements, then the contents of the message element will be displayed to the user, and no further matching will occur.

With the example message mapping file shown above, the error message would be mapped to the message shown in below.



13.3.2 Configuring LDAP Authentication

Section 13.3.1, "External Authentication" provides a general discussion of using external authentication modules with the iDashboards server. Here, instructions are provided for configuring a specific authentication module, which is bundled with the iDashboards server. It is designed to authenticate login credentials against a Directory Server using LDAP (Lightweight Directory Access Protocol), and is referred to as the LDAP authentication module.

Note: Microsoft Active Directory, Novell eDirectory and Oracle Internet Directory are all examples of directory servers.

Note: The information in Section 13.3.1, "External Authentication" should be read and fully understood before attempting to configure the LDAP authentication module. Also, a general knowledge of basic LDAP principles and terminology is recommended.

How It Works

When a user enters a username and password upon login, the LDAP authentication module will attempt to validate those login credentials in one of two configurable ways:

1. In the simplest mode, it will use the login credentials to attempt to log into (or "bind to", in LDAP parlance) the Directory Server. In many cases it will need to decorate the username with other information to form all or part of the user's DN (Distinguished Name) before presenting the credentials to the Directory Server. If the bind is successful, then the user is considered to be authenticated. **This mode of authentication is referred to as "mode 1" authentication.**
2. In some cases, it may not be possible to bind to the Directory Server with a partial DN, or to build the user's full DN prior to the bind attempt. In these cases, the LDAP authentication module must be configured to first search for the user's entry in the directory to determine the DN, and then bind to the directory with the DN and password.

   In order to search for the user's entry, however, the LDAP authentication module must first bind to the Directory Server. This may be accomplished by using the DN and password of another user with bind and search privileges, that are passed to the LDAP authentication module as configuration options, or by binding anonymously (without any username or password) and searching the directory, if the Directory Server allows it. **This mode of authentication is referred to below as "mode 2" authentication.**

In both modes, authentication is accomplished by binding to the directory with the user's login credentials. Therefore, it is absolutely essential that users have bind privileges in the Directory Server in order for the LDAP authentication module to work.

13.3.2.1 Installing the LDAP Authentication Module

Because the LDAP authentication module is built into the iDashboards server, there is no need to add its LoginModule class to the application server's classpath. To make iDashboards use it, the following line should be added to the ivizgroup.properties file, replacing any current setting for "login.module":

```
login.module=com.ivizgroup.idb.security.auth.LDAPLoginModule
```

13.3.2.2 Configuring the LDAP Authentication Module

Simply installing the LDAP Authentication Module is not sufficient for it to work properly; it must also be configured. This is done by setting its configuration options in the ivizgroup.properties file. As discussed in Section 13.3.1, "External Authentication", the name of each configuration option is prefixed with "login.module." in the ivizgroup.properties file. Therefore, if the option is documented below to have the name "debug", with possible values of "true" or "false", a "true" entry in the ivizgroup.properties file would look like:

```
login.module.debug=true
```

The configuration options for the LDAP authentication module are as follows:

authMode - (Optional, default is 1) This is the numerical identifier of the authentication mode to be used, as described in the section "How it Works" above.

connectionURL - (Required) This is the URL of the Directory Server. Normally, it will be of the form:

```
ldap://<hostname>:<port number>
```

Where <hostname> is the name of the host on which the directory service is running, and <port number> is the port on which it accepts incoming connections. (The standard LDAP port is 389.)

userBindPattern - (Mode 1 only, optional) This option indicates what decoration, if any, should be applied to the username before attempting to bind to the directory with it. If it is used, it should be a string containing the substring "{0}" (zero enclosed in curly braces.) The "{0}" substring will be replaced with the username prior to the bind attempt.

An example of this option might be for a full DN, for example:

```
cn={0},ou=Managers,dc=mycompany,dc=com
```

With the example above, if a user attempts to login with the username "jsmith", then the DN that would be used for the bind attempt would be:

```
cn=jsmith,ou=Managers,dc=mycompany,dc=com
```

Some directory servers might allow a user to bind using only part of a DN, for example "cn=jsmith". In such cases, the userBindPattern would be:

```
cn={0}
```

If the userBindPattern is not provided for a mode 1 authentication, then an attempt will be made to bind to the directory with only the username provided by the user.

securityLevel - (Optional) This option indicates the level of security that will be used when communicating with the directory server. Allowable values are "none", "simple" and "strong". If omitted, the level will be determined by the service provider.

connectionName - (Optional for mode 2, ignored for mode 1) This is the username used to bind to the directory server when searching for the login user. It may be a simple username, for example "netadmin", or it may be a partial DN, for example, "cn=netadmin", or it may be a full DN. If it is not provided, then an anonymous bind will be attempted when searching for the user; therefore, the server must be configured to allow anonymous binds.

connectionPassword - (Optional for mode 2, ignored for mode 1) If a value is provided for the connectionName option, this option must indicate the password that will be used to bind to the directory when searching for the login user. The password may be in cleartext, or it may be obfuscated with the idb_encrypt tool described in Section 16.1, "Using the idb_encrypt tool". If it is obfuscated, then the connectionPassword.encrypted option (described below) must be set to "true".

connectionPassword.encrypted - (Optional for mode 2, ignored for mode 1) If the connectionPassword option (described above) is obfuscated with the idb_encrypt tool, then this option must be set to "true". If the connectionPassword option is blank or in cleartext, then this option can be set to "false" or left blank.

userSearch - (Required for mode 2, ignored for mode 1) The value of this option is a string representing the LDAP search filter that will be used to search the directory for the login user. It must contain the substring "{0}" (zero enclosed in curly braces.) The "{0}" substring will be replaced with the username prior to the search. A typical example might be:

```
(cn={0})
```

Note: See http://support.idashboards.com/links/ldapsearchfilter for more information on LDAP search filters.

userBase - (Optional for mode 2, ignored for mode 1) This is the DN of the base element in the directory that is used for user searches. If it is blank, the top-level element of the directory context will be used; therefore, providing a value for this option may speed up user searches.

userSubtree - (Optional for mode 2) The value of this option must be "true" or "false". (Default, if omitted, is "true"). If "true", then user searches will encompass the entire subtree rooted at the element specified by the userBase option. If it is false, then only the first level below the element specified by the userBase option will be searched.

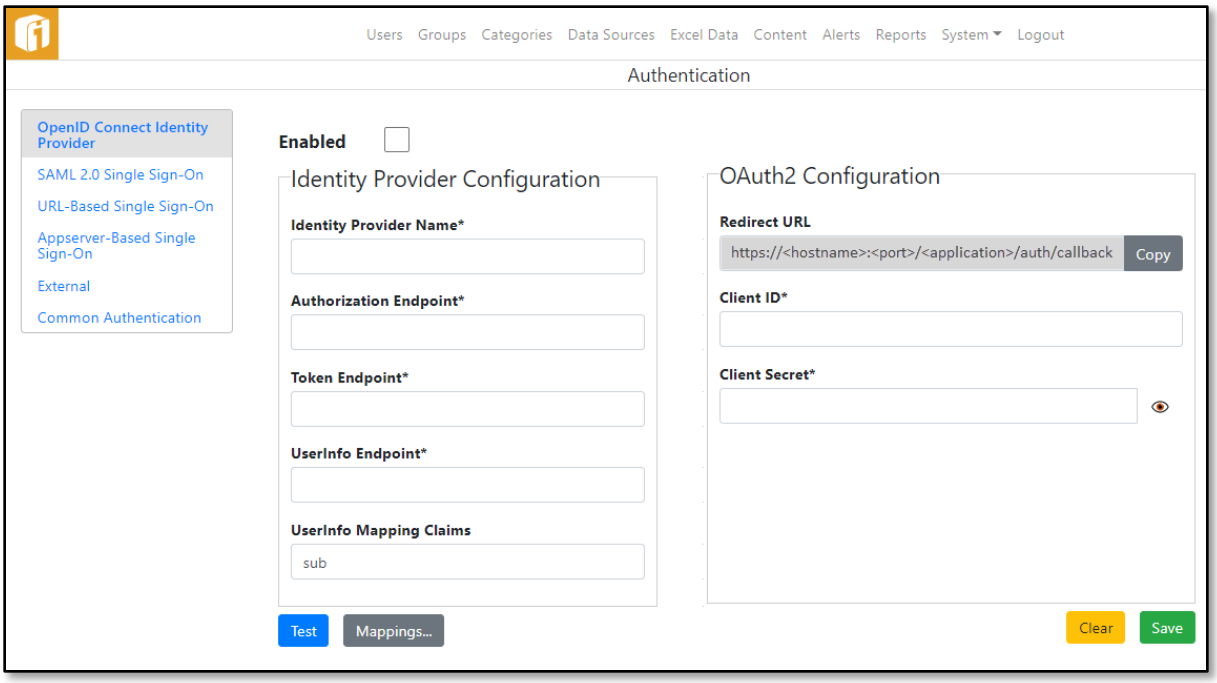
referral - (Mode 2 only, optional) If a user search is conducted on a directory tree spanning multiple servers, this setting determines whether only the main server (the one indicated by the connectionURL option) is searched, or other servers linked through "referrals" are searched also. Allowable values are "follow", which indicates that linked servers should also be searched, and "ignore", which indicates that linked servers should not be searched. The default is "follow".

remoteServerBind - (Mode 2 only, optional) If a user's record is found on a server other than the main server (the one indicated by the connectionURL option), a value of "true" for this setting means the user should be authenticated by binding to the server where the record was found (the remote server) with the user's credentials, and a value of "false" indicates that the authenticating bind should be to the main server. The default is "true", since the user is more likely to have bind privileges on the (remote) server where the record was found.

13.3.3 OpenID Connect Identity Provider

iDashboards supports the ability to configure an external identity provider to manage user authentication. An OpenID Connect (OIDC) identity provider is an identity layer on top of OAuth2. Many social platforms offer OIDC identity providers, including Google, Facebook and Twitter. Commercial OIDC identity providers including Okta, Auth0 and OneLogin.

A "LOG IN WITH <name>" button will appear on the initial login page when authentication is enabled. When a user clicks this button, authentication will be 'handed-off' to the identity provider to establish the identity of the user. Once the identity provider has established the user's identity; their identity information will be supplied to iDashboards.



To configure an OIDC identity provider, generally an OAuth 2.0 application is created with the identity provider to supply it with information about iDashboards. Information is then supplied by the identity provider that is used to configure the identity provider within iDashboards.

13.3.3.1 Configuring the Identity Provider

The **Identity Provider Name** is simply a name given to the configuration that appears on the "LOG IN WITH" button on the initial login page when authentication is enabled.

The identity provider endpoints (**Authorization Endpoint**, **Token Endpoint**, **UserInfo Endpoint)** are supplied by the identity provider and are generally not specific to the iDashboards application.

User Mapping Claims

The information returned by the identity provider describing the user are called claims. This information is how users from the identity provider are mapped to iDashboards users. The claims that are most likely to uniquely identify a user are:

sub – Subject – Identifier for the user supplied by the identity provider. This claim is required because when a user authenticates themselves in the User Settings this mapping is populated.

email – The user's email address as supplied to the identity provider to establish the user's identity.

There are other claims that may be returned by the identity provider, but generally they are not sufficient to uniquely identify a user. Additional claims can be listed to be used to

establish the user mapping. ALL of the information specified in the mapping will be used to map the identity provider user to an iDashboards user.

13.3.3.2 OAuth2 Configuration

The client ID and client secret are OAuth2 credentials and are provided by the identity provider and are associated specifically with iDashboards. They are generated by the identity provider to identify requests from iDashboards.

The **Redirect URL** is supplied to the identity provider to indicate where the user information should be returned to after the user has authenticated with the identity provider. This URL is typically required when configuring the iDashboards application with the identity provider before the client ID and client secret are generated.

13.3.3.3 User Authentication Mappings

Once the user's identity has been established by the identity provider (usually in the form of an identity-provider specific login screen), the iDashboards user that is associated with the authenticated user must be established. This association is determined using authentication mappings. To associate an authenticated user with an iDashboards user, the authentication mappings are examined to determine a matching iDashboards user. This will then be the iDashboards user that will be associated with the user that has been authenticated by the identity provider.

Users may establish their own authentication mapping via the User Settings in the Viewer or the Builder. Once they have established the mapping via the Settings, the 'sub' mapping claim will be populated because this is the only claim that is required from the identity provider and it uniquely identifies the user.

13.3.3.4 Single Sign-On

A special URL can be used to bypass the login screen and immediately contact the identity provider for authentication. If the URL to access the application is:

```
http://dashboard.mycompany.com/idashboards/
```

Then the following URL will automatically log the user into iDashboards:

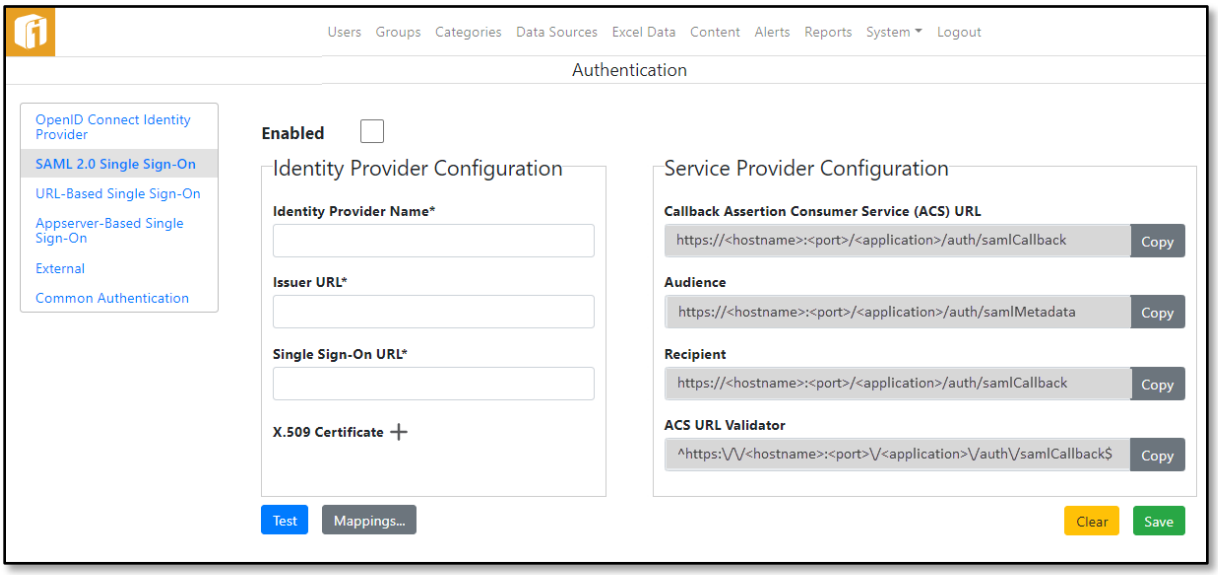
```
http://dashboard.mycompany.com/idashboards/?sso=$auth=oidc
```

When iDashboards is invoked with this type of URL, the user is authenticated with the identity provider. The associated iDashboards user is then determined based on the authentication mappings and the identity information returned by the identity provider.

13.3.4 SAML 2.0 Single Sign-On

Similar to OpenID Connect, iDashboards supports the ability to configure an external identity provider to manage user authentication with a Security Assertion Markup Language 2.0 (SAML) identity provider. OIDC has become the more popular implementation choice for communicating with an identity provider and most commercial identity providers provide both an OIDC and SAML interface. Commercial SAML identity providers including Shibboleth, Okta, Auth0 and OneLogin.

A "LOG IN WITH <name>" button will appear on the initial login page when authentication is enabled. When a user clicks this button, authentication will be 'handed-off' to the identity provider to establish the identity of the user. Once the identity provider has established the user's identity; their identity information will be supplied to iDashboards.



In SAML 2.0 terminology, the entity requesting authentication from the identity provider is called the service provider. In this case, iDashboards is the service provider. As with OIDC, an application is generally created with the identity provider and information about iDashboards is provided to the identity provider and information about the identity provider is used to configure SAML authentication in iDashboards.

13.3.4.1 Configuring the Identity Provider

The **Identity Provider Name** is simply a name given to the configuration that appears on the "LOG IN WITH" button on the initial login page when authentication is enabled.

The following URLs are obtained from the identity provider and as specifically associated with the iDashboards application.

Issuer URL – This is the entity ID of the identity provider.

Single Sign-On URL – This is the URL to which iDashboards issues its initial authentication request. It is sometimes referred to as the login URL.

X.509 Certificate – Most identity providers require authentication requests to be signed using an X.509 certificate. The signing algorithm that is used to sign the requests is also provided by the identity provider.

13.3.4.2 Service Provider Configuration

iDashboards is considered the service provider when configuring the identity provider. iDashboards provides the following values to be supplied to the identity provider:

Callback Assertion Consumer Service (ACS) URL – This is the URL to which the identity provider sends identity information after the user has been authenticated. It is sometimes referred to as the Application Callback URL or Single Sign-On URL.

Audience – This is entity ID of the service provider.

Recipient – For iDashboards, this is just the ACS URL.

ACS URL Validator – This is a regular expression used by some identity providers to ensure that responses are issued to the correct URL.

iDashboards uses the emailAddress NameIDFormat, which indicates the information returned by the identity provider to identify a user.

13.3.4.3 User Authentication Mappings

Once the user's identity has been established by the identity provider (usually in the form of an identity-provider specific login screen), the iDashboards user that is associated with the authenticated user must be established. This association is determined using authentication mappings. To associate an authenticated user with an iDashboards user, the authentication mappings are examined to determine a matching iDashboards user based on the email address returned by the identity provider. This will then be the iDashboards user that will be associated with the user that has been authenticated by the identity provider.

Users may establish their own authentication mapping via the User Settings in the Viewer or the Builder. Once they have established the mapping via the Settings, the email address provided by the identity provider will be populated in the mapping associated with the user.

13.3.4.4 Single Sign-On

A special URL can be used to bypass the login screen and immediately contact the identity provider for authentication. If the URL to access the application is:

```
http://dashboard.mycompany.com/idashboards/
```

Then the following URL will automatically log the user into iDashboards:

```
http://dashboard.mycompany.com/idashboards/?sso=$auth=saml
```

When iDashboards is invoked with this type of URL, the user is authenticated with the identity provider. The associated iDashboards user is then determined based on the authentication mappings and the identity information returned by the identity provider.

13.3.5 URL-Based Single Sign-on

The iDashboards server can be configured to allow single sign-on (referred to here as SSO) to the iDashboards application. What this means is that a user can log into a company intranet through a web browser, and, from a link on an intranet web page, access the iDashboards application without supplying additional login credentials.

iDashboards SSO is an advanced feature, which in most cases must be implemented by a web developer, a system administrator, or both.

There are two basic types of SSO available in iDashboards, URL-based SSO, which is discussed in this chapter, and Appserver-based SSO, which is discussed in Chapter 13.3.6, "Appserver-Based Single Sign-on". Both types of SSO can be enabled simultaneously.

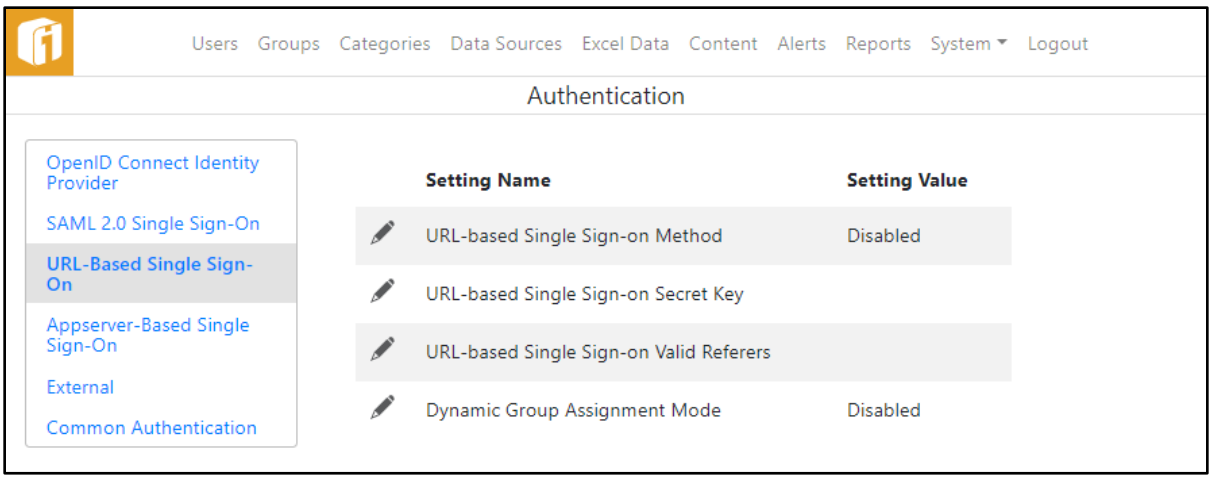
CAUTION

There are certain security risks inherent in URL-based Single Sign-on, whereby an attacker could gain unauthorized access to the iDashboards application. It cannot be made completely invulnerable to a determined and resourceful attacker. The known security risks are explained in this document. They should be thoroughly understood and evaluated before enabling URL-based Single Sign-on. Other unknown or undocumented vulnerabilities may also exist.

With URL-based SSO, the URL used to access iDashboards in SSO mode contains encoded information that is used to verify, to a degree, that the SSO login request is legitimate. In virtually all cases this URL must be constructed dynamically through a programming language such as Java, ASP, PHP, Visual Basic or C#, as opposed to being embedded in a static HTML page, because at minimum it must contain the iDashboards username of the user. There are several types of URLs that can be used for URL-based SSO, each of which is associated with a URL-based SSO "method" which the iDashboards server must be configured to accept.

13.3.5.1 Enabling URL-Based SSO

URL-based SSO is disabled by default. It is enabled by selecting an SSO method on the System Settings screen of the iDashboards application. To enable it, go to the System Settings screen and select "URL-Based Single Sign-On" from the Setting Category list, then select the Edit icon ( ) for the system setting "URL-based Single Sign-on Method". From the dropdown that appears in the modification form, select the SSO method (other than "Disabled") that will be used (Referer Check, Non-Expiring, Expiring, Secret Key, Password).



The SSO method that is selected will dictate the type of URL that must be constructed to accomplish single sign-on. Each method is explained in the following chapters; however there are several key concepts that relate to all methods that will be explained first.

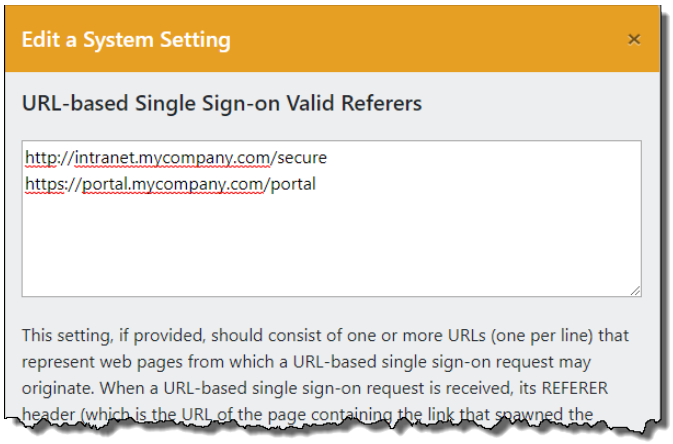
> *Note: See Chapter 13, "System Configuration" for more information on modifying general system settings.*

13.3.5.2 KEY CONCEPT: Referer Checks

An HTTP request, sent from a browser to a web server or application server, contains bits of information in addition to the URL that are called "headers." When a user selects a link on a web page, the request sent to the server contains a header named "referer" that is the URL of the page containing the link. Regardless of the SSO method used, iDashboards SSO can be configured to only allow SSO logins that originate from a predefined list of "referring" web pages, identified by their URLs. If such a list has been configured, iDashboards will check the referer header of an SSO login request against the list, and if the referer URL *begins with* one of the valid referer URLs, then the login is allowed to proceed, provided all other security checks pass.

The list of valid referer URLs are stored as an iDashboards system setting. To configure the list, edit the system setting "URL-based Single-Sign-on Valid Referers." In the modification dialog, enter the list of valid referring URLs, one-per-line and select the "Save" button. The total, combined length for entries in the list cannot exceed 500 characters, including end-of-line characters. Also, it should be noted, URLs that begin with "https" can be problematic, since some browsers, under some circumstances, will not include referer headers when a link is selected on a page loaded from an HTTPS request.

> *Note: Although the proper spelling is "referer", the actual HTTP header name is misspelled as "referer" by the HTTP/1.1 Specification. The spelling used in this document is consistent with that used in the HTTP/1.1 Specification.*
>
> *Note: HTTPS stands for "HTTP over SSL (Secure Socket Layer), and is a protocol used to pass information between a web server and a browser in encrypted format.*



Note that a partial, case-sensitive match is used when checking a referring URL against the list of valid referring URLs. For example, if the list *only* contains:

```
http://intranet.mycompany.com/secure
```

Then the following referring URLs will pass the check:

```
http://intranet.mycompany.com/secure/usermenu.jsp

http://intranet.mycompany.com/secure/marketing/outlook.asp
```

And the following URLs will not, because the underlined portions do not match:

```
https://intranet.mycompany.com/secure/usermenu.jsp

http://intranet.mycompany.com/index.jsp

http://intranet.mycompany.com/SECURE/usermenu.jsp
```

Regardless of the SSO method used, if a list of valid referers is configured, the referer header of an SSO login request must match one of the valid referers in order to succeed. Therefore, referer checks can add an additional measure of security and should be used when possible.

VULNERABILITIES

Web browsers typically do not allow a user to add or modify HTTP request headers; however, a resourceful attacker could use specialized tools to "spoof" a referer header in such a way that an illicit SSO login attempt would pass a referer check.

13.3.5.3 KEY CONCEPT: Obfuscation

Several of the SSO methods described below require that part of the SSO URL be obfuscated. This means it is encoded in such a way that the information it contains is made unreadable by the browser user. Obfuscation provides a measure of security in the sense

that a casual attacker presumably cannot create a properly obfuscated URL containing the needed information for an SSO login.

When the page containing the SSO URL is generated by a Java Servlet or JSP, obfuscation is done by a Java library provided with iDashboards. The library is contained in the file idb_encrypt.jar, which is in the tools directory of the iDashboards installation CD. It must be available in the classpath of any Java code that uses it.

The following snippet of Java code shows how a string is obfuscated:

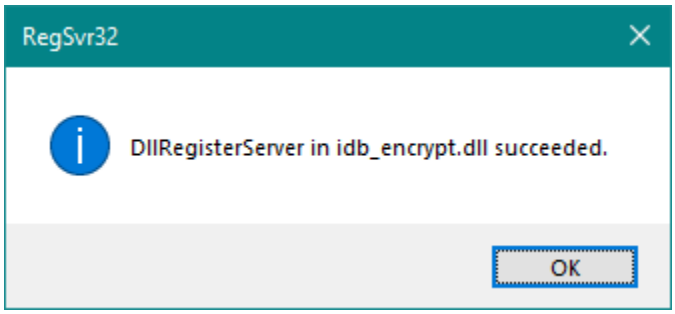
```
String cleartext = "abcdefg";

String obfuscated = com.ivizgroup.util.Codec.obfuscate(cleartext);
```

When the page containing the SSO URL is generated by a Windows-based technology such as Microsoft ASP, the obfuscation can be done by an ActiveX DLL provided with iDashboards. The DLL file is named idb_encrypt.dll, and it is also in the tools directory of the iDashboards installation CD. Before it can be used, it must be installed and registered on the server where the URL obfuscation will take place. This is typically done from a command prompt with the Windows utility regsvr32.exe. In the directory where idb_encrypt.dll is located, type the command:

```
regsvr32 idb_encrypt.dll
```

If the registration succeeds, a message box will appear.



After idb_encrypt.dll has been registered, the following snippet of VBScript, embedded in an ASP page, should successfully obfuscate the string named "cleartext":

```
Dim myObj, cleartext, obfuscated

Set myObj = server.createObject("idb_encrypt.Codec")

cleartext = "abcde"

obfuscated = myObj.obfuscate(CStr(cleartext))
```

An obfuscated string will always be twice the length of its cleartext counterpart.

VULNERABILITIES

The iDashboards obfuscation library uses a proprietary, weak two-way encryption algorithm to obfuscate text. It is possible that a resourceful attacker could reverse-engineer the algorithm and use it to obfuscate SSO login URLs, or decode obfuscated ones. Moreover, anyone with access to the idb_encrypt.jar file or idb_encrypt.dll, and the proper knowledge, could use it to obfuscate SSO login URLs.

13.3.5.4 KEY CONCEPT: SSO URLs

An SSO URL consists of two parts: The URL of the iDashboards application and an additional "sso parameter." If the URL to access the iDashboards application is:

```
http://dashboard.mycompany.com/idashboards/
```

Then an SSO URL would look like this:

```
http://dashboard.mycompany.com/idashboards/?sso=<sso parameter value>
```

with <sso parameter value> replaced by a string of text, usually obfuscated, that contains the information needed for the SSO login. This URL structure is common to all of the SSO methods described below; what differs among them is the sso parameter value. Therefore, the chapters following that discuss the different SSO methods will explain only how to construct the string used as the sso parameter value in an SSO URL.

13.3.5.5 SSO Method: Referer Check

The Referer Check method is one of two iDashboards SSO methods that do not require the sso parameter value to be obfuscated. For this reason, it is a practical choice when obfuscating the URL with one of the provided libraries is difficult or impossible. It is also the only method that *requires* a referer check. If there are no valid referer URLs configured, then all SSO logins will fail with this method.

For the Referer Check method, the value of the sso parameter consists solely of the iDashboards username of the user. As an example, for the user "janesmith" an SSO URL would look like this:

```
http://dashboard.mycompany.com/idashboards/?sso=janesmith
```

VULNERABILITIES

Since the Referer Check method relies solely on referer checks for security, the vulnerabilities described previously for referer checks would apply.

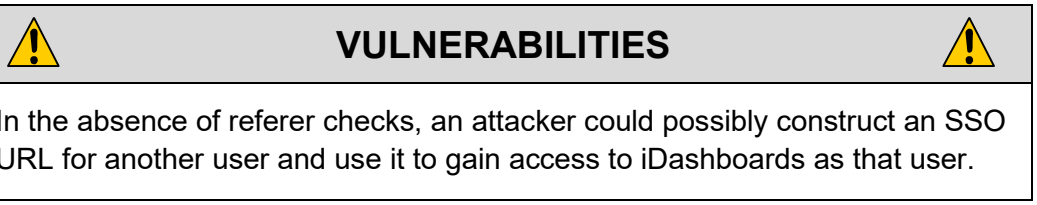
13.3.5.6 SSO Method: Non-Expiring

This method is similar to the Referer Check method, in that the only information needed in the sso parameter value is the iDashboards username of the user. It is different in two ways, however:

1. It doesn't require any referer checks, although referer checks will be applied if valid referer URLs are configured.
2. The username must be obfuscated. So for a user "janesmith", an example SSO URL would have as its sso parameter value "6b636a6d634d29f469", which is the obfuscated form of "janesmith". For example::

```
http://dashboard.mycompany.com/idashboards/?sso=6b636a6d634d29f469
```

This method is referred to as "non-expiring" because the SSO URL can be reused any number of times as long as the iDashboards username remains valid. If referer checks are not also used it can be bookmarked, or shared with other users, or linked from any web page. Therefore, it is strongly recommended that this method be used only in conjunction with referer checks. If referer checks are used, the SSO URL can be reused, but only from pages that match one of the valid referring URLs. It will be as secure as the Referer Check method, with the possible additional measure of security provided by the obfuscation of the username.



13.3.5.7 SSO Method: Expiring

When the Expiring SSO method is used, the sso parameter value contains the username and an expiration date and time, beyond which the SSO URL will not work. An example of the sso parameter value for this SSO method, in unencrypted form, would be:

```
janesmith|expires=2015-09-05 15:43:21
```

Obfuscated, the above string would be:

```
6b636a6d634d29f4697e6170604932e5723f363820156db1332f3538301175ba
```

An example URL would be (combined on one line):

```
http://dashboard.mycompany.com/idashboards/?sso=
6b636a6d634d29f4697e6170604932e5723f363820156db1332f3538301175ba
```

In unobfuscated form, the sso parameter value consists of two fields separated by a pipe symbol. The first field is the username, and the second field is of the form:

```
expires=yyyy-MM-dd HH:mm:ss
```

with "yyyy-MM-dd HH:mm:ss" being the expiration date and where:

- **yyyy** is the four-digit year
- **MM** is the two-digit month
- **dd** is the two-digit day of the month
- **HH** is the two-digit hour of the day, in 24 hour format. (For example, 12:00 a.m. would be 00 and 11:00 p.m. would be 23.)
- **mm** is the two-digit minute of the hour
- **ss** is the two-digit second of the minute

The advantage that the Expiring method has over non-expiring ones is that a URL has a finite lifespan, and will not work beyond its expiration timestamp. This can effectively limit a user's ability to bookmark or share an SSO URL.

There are several practical considerations involved, however. For instance, how far into the future should the expiration timestamp be? If it's only a few minutes into the future, then a legitimate user might select an SSO link that's gotten stale and be denied what is otherwise a legitimate SSO login. If it's days into the future, the URL might be misused before it expires.

The most effective way to use an expiring SSO URL is to not embed it directly in a page, but rather return it in the "Location" header of an HTTP response, along with a 302 (Moved Temporarily) response code. In this case, the expiration time can be a mere seconds into the future. It will effectively only be good for a single login. In a Java Servlet, this is accomplished by calling the sendRedirect method of the HttpServletResponse object. In the Java code snippet below, "ssoUrl" is a String variable representing the SSO URL, and "response" is the HttpServletResponse object:

```
response.sendRedirect(ssoUrl);
```

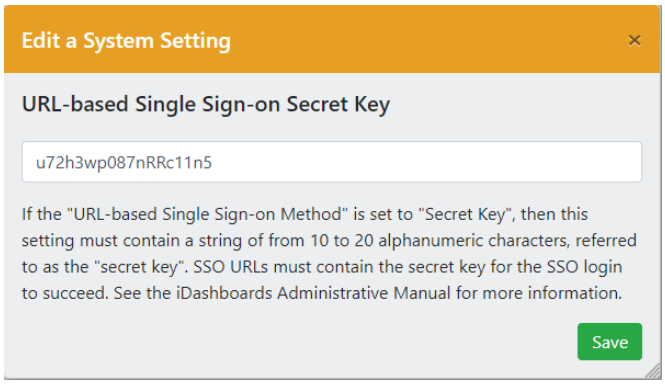
By using a redirect scheme, the SSO URL is constructed on-the-fly *after* the user selects the link to access iDashboards.

VULNERABILITIES

The vulnerabilities associated with obfuscation apply here; an attacker with the ability to obfuscate text using the iDashboards obfuscation algorithm could construct an SSO URL for another user, which expires well into the future, and use it to gain access to iDashboards as that user. Therefore, referer checks should also be used in conjunction with this SSO method when practical. Also, if a redirect URL (that returns an HTTP 302 response code, described above) is used, it could be exploited by an attacker to gain access to iDashboards as another user. Therefore, the code that generates the 302 response should do its own referer check, and other checks that authenticate the user, to verify the authenticity of the request.

13.3.5.8 SSO Method: Secret Key

With this method, the SSO parameter value must contain a "secret key" in order for an SSO login to succeed. The secret key is a string of from 10 to 20 alphanumeric characters, which is stored as an iDashboards system setting in clear-text. To configure the secret key, open the System Settings screen of the Administrator application and edit the setting "URL-based Single-Sign-on Secret Key." In the modification form, enter a random string of alphanumeric characters and select the "Save" button.



The SSO parameter value used with the Secret Key method contains the username and the secret key. An example of the SSO parameter value, in unencrypted form, would be:

```
janesmith|key=u72h3wp087nRRc11n5
```

Notice that the username is followed by a pipe symbol, followed by "key=", followed by the secret key. The entire SSO parameter value must be obfuscated; the obfuscated form of the parameter value above would be:

```
6b636a6d634d29f4697e6f6d691d35b75b6a377f601078b76f50566b21112eb5
```

An example URL would be (combined on one line):

```
http://dashboard.mycompany.com/idashboards/?sso=
6b636a6d634d29f4697e6f6d691d35b75b6a377f601078b76f50566b21112eb5
```

VULNERABILITIES

As mentioned previously, the secret key used with the Secret Key SSO method is stored in the iDashboards repository database in cleartext. It is also displayed in cleartext on the System Settings screen. An attacker with access to the repository database or the Administrator application could get the value of the secret key, and could possibly construct an illicit SSO URL, if he or she also had the ability to obfuscate the SSO parameter value. Therefore, referer checks should be used in combination with the Secret Key method.

13.3.5.9 SSO Method: Password

The Password method is the only SSO method that requires knowledge of the user's iDashboards password, since it must be included in the obfuscated SSO parameter value. iDashboards uses the username and password to authenticate the SSO request, and if the authentication is successful, the user is granted access to iDashboards.

A typical SSO parameter value used with the Password SSO method would look like this:

```
janesmith|pw=m3j@n3
```

In the above example, the username "janesmith" is followed by a pipe symbol, which is followed by "pw=", which in turn is followed by janesmith's iDashboards password. (The password must not contain a pipe symbol.) The SSO parameter value must be obfuscated. The above parameter value, in obfuscated form, would be:

```
6b636a6d634d29f4697e747f2d4d73ea416c37
```

An example URL would be (combined on one line):

```
http://dashboard.mycompany.com/idashboards/?sso=
6b636a6d634d29f4697e747f2d4d73ea416c37
```

Just-In-Time Obfuscation: It was stated above that the Password SSO method requires obfuscation of the SSO parameter value. This is true; however, iDashboards provides a way of creating a properly obfuscated SSO URL from a cleartext URL, which makes this SSO method practical even when obfuscation is impractical. The base URL is the same URL used to access iDashboards, with "/login" appended. So if the URL used to access iDashboards is:

```
http://dashboard.mycompany.com/idashboards/
```

The base URL for just-in-time obfuscation would be:

```
http://dashboard.mycompany.com/idashboards/login
```

The username and password must be included as parameters to the URL, named "user" and "password" respectively. So combining the above base URL, username and password, an un-obfuscated URL for Password SSO would be:

```
http://dashboard.mycompany.com/idashboards/login?user=
janesmith&password=m3j@ne
```

When the above URL is invoked, the iDashboards server will redirect the user's browser to the following SSO URL:

```
http://dashboard.mycompany.com/idashboards/?sso=
6b636a6d634d29f4697e747f2d4d73ea416c61
```

A dashID parameter can be included in the unobfuscated URL to autoload a dashboard:

```
http://dashboard.mycompany.com/idashboards/login?user=
janesmith&password=m3j@ne&dashID=42
```

See Section 13.3.5.10, "KEY CONCEPT: Autoloading Dashboards" for more information on autoloading dashboards.

VULNERABILITIES

When the Password SSO method is employed, it is possible that a user's password might be cached in a user's browser history list in cleartext or obfuscated form, making it susceptible to discovery by an attacker. It may also be stored in webserver or application server access logs. This risk can be mitigated by using the Password SSO method in conjunction with HTTP POST requests, where the parameters are sent in the body of the request rather than as part of the URL.

13.3.5.10 KEY CONCEPT: Autoloading Dashboards

With all of the SSO methods described above, the SSO parameter value can be constructed in such a way that a specified dashboard will automatically load upon a successful SSO login, provided that the user's group has access to that dashboard. This is accomplished by including the dashboard ID number in the SSO parameter value. It can be appended to the un-obfuscated SSO parameter value as shown here:

```
janesmith|key=u7Zh3wp087nRRc11n5|dashID=323
```

In the example above, the normal SSO parameter value is followed by a pipe symbol, which is followed by "dashID=" (case-sensitive), which in turn is followed by the dashboard ID number.

The dashboard ID number is the primary key used to identify a dashboard in the iDashboards repository database. The dashboard ID for an individual dashboard is displayed on its Dashboard Features dialog in the iDashboards application. Dashboard IDs for multiple dashboards may also be read directly from the repository database, using an ISQL or reporting tool. The following query will produce a list of dashboards, including their dashboard ID numbers (the dash_id column), names and descriptions, as well as their associated categories:

```
SELECT d.dash_id, d.dashname, d.dashdescr, c.category_id,
c.category_descr

FROM fv_dashboard d, fv_dashcategory c

WHERE d.category_id = c.category_id

AND c.category_id <> 1
```

The condition "c.category_id <> 1" will exclude dashboards that are associated with users' Personal categories.

13.3.5.11 KEY CONCEPT: Dynamic Group Assignment

Users and Securities explain how each iDashboards user belongs to one or more user groups, and how the privileges of those groups determine, to a large extent, the user's privileges in the iDashboards system. Normally, a user's groups are set through the user management screens that are discussed in the Users and Securities sections. At the time of a URL-based SSO login, however, a user can be dynamically assigned a single group, and that group assignment will override the user's normal group settings for the duration of that login session. During the login session, the permissions of the dynamically assigned group will determine the permissions of the user.

Enabling Dynamic Group Assignment: Dynamic group assignment is disabled by default. To enable it, go to the Settings screen and select "URL-Based Single Sign-On" from the Setting Category dropdown, then select the Edit icon ( ) for the system setting "Dynamic Group Assignment Mode." The four possible choices for this setting are:

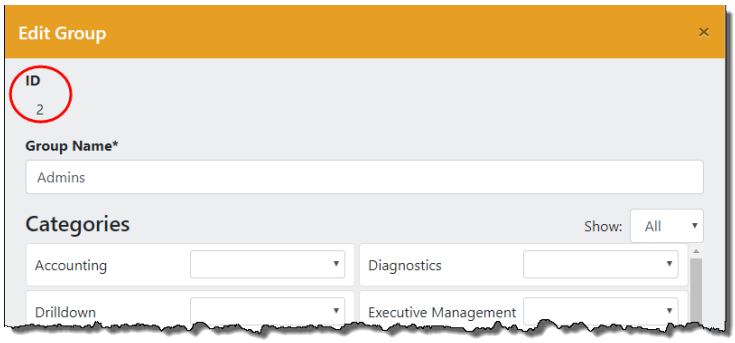
- **Disabled** — Dynamic group assignment is disabled, and any attempts at dynamic group assignment during SSO login will result in an error.
- **Viewer Role** — When a user's group is dynamically assigned during SSO login, the user will also be assigned the "Viewer" role for the duration of the login session, which means the user will not be able to make or save any changes to charts or dashboards, and will not have access to a Personal Category.
- **View-Only Category Access** — When a user's group is dynamically assigned, the user will have view-only access to each of the group's assigned categories, regardless of the group's actual level of access to those categories.
- **Full Access** — When a user's group is dynamically assigned, the user's Category access privileges will be those of the assigned group.

Adding Dynamic Group Assignment to SSO: With any of the SSO methods described above, the group can be dynamically assigned by including the groups ID number in the SSO parameter value. It can be appended to the unobfuscated sso parameter value as shown here:

```
janesmith|key=u7Zh3wp087nRRc11n5|gid=5
```

In the example above, the normal SSO parameter value is followed by a pipe symbol, which is followed by "gid=" (case-sensitive), which in turn is followed by the group ID number.

The group ID is the number which uniquely identifies a group in the iDashboards system. While a group's name may be changed, its ID number always remains the same. A group's ID number is visible the "Edit Group" screen which is accessed by selecting the "GROUPS" tile from the Administrator home screen, or "Groups" on the menu bar, and then the group's "Edit" icon ( ).



13.3.5.12 Troubleshooting

If an SSO login fails due to a malformed SSO parameter value, the error message returned to the user will not provide details about why the parameter value was invalid. The error message will, however, include a numeric code in parentheses following the error message indicating the cause of the failure. The error codes and their meanings are as follows:

10 The sso parameter was missing or empty.

20 The sso parameter value was malformed. Recall that an sso parameter value is a series of fields separated by pipe symbols. The first field is always the username, and the following fields should be in "name=value" format. An error code of 20 indicates that one of the fields following the username was not in this format.

30 The sso parameter value was supposed to be obfuscated and it was not.

40 One of the fields following the username field contained either an invalid name or invalid value.

100 The referer check failed.

200 The expiration timestamp was missing or not properly formatted. Check that it was prefixed with "|expires=".

250 The expiration timestamp was malformed. Check that it conforms to the format described in Section 13.3.5.7, "SSO Method: Expiring".

300 The secret key was needed but was missing from the sso parameter value.

310 The secret key was needed but was not configured in the database.

350 The secret key included in the sso parameter value did not match the one configured in the database.

400 A group ID was included in the sso parameter value, but dynamic group assignment was disabled.

410 The group ID included in the sso parameter value was not an integer.

420 The group ID included in the sso parameter value was not an actual group ID.

490 A database or other system error occurred.

900 The SSO method configured in the database is invalid. This should never occur, but if it does, reset the desired SSO method.

13.3.6 Appserver-Based Single Sign-on

The iDashboards server can be configured to allow single sign-on (referred to here as SSO) to the iDashboards application. What this means is that a user can log into a company intranet through a web browser, and, from a link on an intranet web page, access the iDashboards application without supplying additional login credentials.

iDashboards SSO is an advanced feature, which in most cases must be implemented by a web developer, a system administrator, or both.

There are two basic types of SSO available in iDashboards, URL-based SSO, which is discussed in Section 13.3.3, "OpenID Connect Identity Provider", and SAML 2.0 Single Sign-On in Section 13.3.4, "SAML 2.0 Single Sign-On".

Note: URL-Based Single Sign-On, and Appserver-based SSO, which is discussed in this section, can be enabled simultaneously.

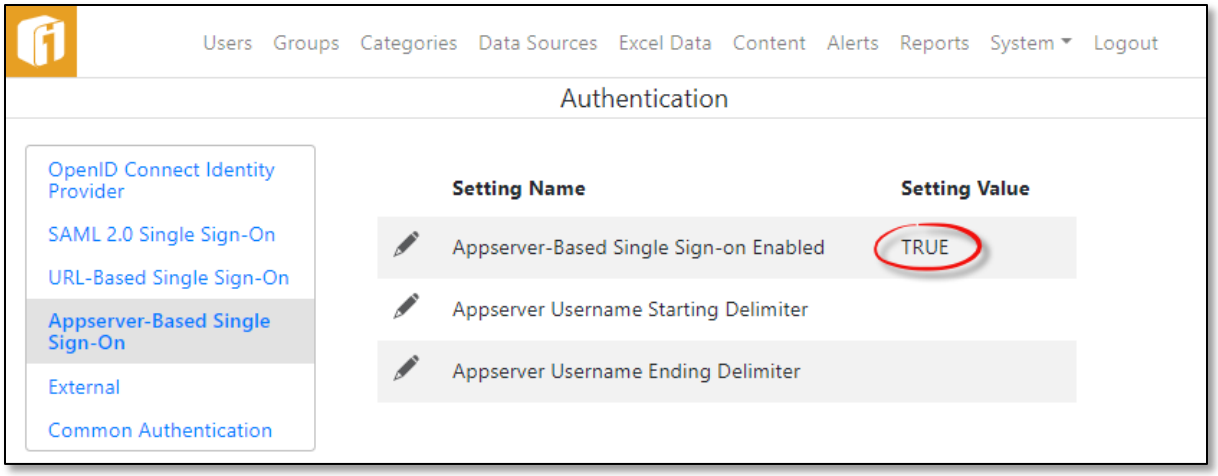
With appserver-based SSO, a user is authenticated by the application server hosting iDashboards before the application is loaded by the user's web browser. When a user attempts to access the iDashboards application through a special URL, the iDashboards server checks to see if she has been authenticated by the application server, and if she has been, she is automatically logged into iDashboards, provided she already has a user account in the iDashboards system. As mentioned repeatedly in Section 13.3.3, "OpenID Connect Identity Provider" and Section 13.3.4, "SAML 2.0 Single Sign-On".

Note: There are certain security weaknesses inherent in URL-Based Single Sign. Appserver-based SSO is much more secure from iDashboards' perspective; for an attacker would have to defeat the authentication mechanism of the application server to gain illicit access to iDashboards, (or to put it another way, appserver-based SSO is no more or no less secure than the application server hosting iDashboards.)

13.3.6.1 Enabling Appserver-Based Single Sign-on

Appserver-based SSO is disabled by default. To enable it, go to the System Settings screen of the Administrator application, and select "Appserver-Based Single Sign-On" from the Setting Category dropdown. Change the system setting named "Appserver-Based Single Sign-on Enabled" from FALSE to TRUE to enable appserver-based SSO.

Note: See Chapter 13, "System Configuration" for more information on modifying general system settings.



13.3.6.2 Accessing iDashboards with Appserver-Based Single Sign-on

As previously mentioned, for appserver-based SSO to work, a user must first be authenticated by the application server that is hosting iDashboards. How this initial authentication occurs depends upon the type of application server being used, and is outside the scope of this document. In most cases the application server will need to be specially configured so that a single authentication is propagated to multiple hosted web applications. Again, configuration of the application server is outside the scope of this document.

When a user has been authenticated to the application server, a special URL can be used to access iDashboards directly, without presenting login credentials. If the URL to access the application is:

```
http://dashboard.mycompany.com/idashboards/
```

Then the following URL will automatically log the user into iDashboards:

```
http://dashboard.mycompany.com/idashboards/?sso=$|auth=container
```

Note that in the above URL, the character after the dollar sign ($) is a pipe symbol (|).

When iDashboards is invoked with this type of URL, the following sequence of events occurs:

1. The iDashboards server will check to see if the user has been authenticated by the application server. If that is not the case, an appropriate error message will be displayed to the user.
2. If the user has been authenticated by the application server, iDashboards will determine the username of the user as he or she is known to the application server. From this username, iDashboards will determine the username of the corresponding user in the iDashboards system. In most cases, the iDashboards username will be the same as the appserver username, but in some cases it may be "derived" from the appserver username, as described in Section 13.3.6.3, "Deriving the iDashboards Username".

3. Once the iDashboards username has been determined, the iDashboards repository is checked to see if a user record exists with that username. If one is found, that user is automatically logged into iDashboards. If a matching user record is not found, an appropriate error message is displayed to the user.

Note: Java Developers: This is done by calling the getRemoteUser() method of the javax.servlet.http.HttpServletRequest class.

13.3.6.3 Deriving the iDashboards Username

In most cases of appserver-based SSO, the iDashboards username should exactly match the username that has been authenticated by the application server. In some cases, however, the appserver username may be of a form that is unsuitable as an iDashboards username; perhaps it is an email address – for example, "joec@mycompany.com" – or it might be a DN (Distinguished Name) used to identify the user in a directory server, for example:

```
dc=mycompany,dc=com,uid=joec,cn=Joseph Carothers,ou=Services
```

In such cases, the iDashboards username can be derived from the appserver username, if it is a substring embedded within the appserver username. This is done by telling iDashboards a substring that will always appear in the appserver username just *before* the iDashboards username, and/or a substring that will always appear in the appserver username just *after* the iDashboards username. These substrings are provided through the system settings "Appserver Username Starting Delimiter" and "Appserver Username Ending Delimiter" respectively, which are in the "Appserver-Based Single Sign-on" setting category.

Suppose that the appserver username is always in the form of an email address, and the corresponding iDashboards username is always the part of the email address prior to the "@" sign. By setting the value of "Appserver Username Ending Delimiter" to "@", and leaving the value of "Appserver Username Starting Delimiter" blank, iDashboards will extract everything from the beginning of the appserver username up to but not including the "@" sign, and use it as the iDashboards username. So if the appserver username is "joec@mycompany.com", the iDashboards username will be "joec".

Suppose now that the username is a DN such as:

```
dc=mycompany,dc=com,uid=joec,cn=Joseph Carothers,ou=Services
```

and the iDashboards username is the value in the "uid" field of the DN. By setting the value of the "Appserver Username Starting Delimiter" setting to "uid=" and the value of "Appserver Username Ending Delimiter" to "," (comma), iDashboards will extract from the appserver username everything between the first occurrence of "uid=" and the closest following occurrence of "," and use it as the iDashboards username. In the above example, that would be "joec".

13.3.6.4 Autoloading Dashboards

The URL used for appserver-based SSO can be modified so that a specified dashboard will load upon a successful SSO login, provided that one of the user's groups has access to that

dashboard. This is accomplished by including the dashboard ID number in the URL as shown below:

```
http://dashboard.mycompany.com/idashboards/?sso=$|auth=container|dashID=323
```

In the example above, the normal SSO URL is followed by a pipe symbol, which is followed by "dashID=" (case-sensitive), which in turn is followed by the dashboard ID number.

The dashboard ID number is the primary key used to identify a dashboard in the iDashboards repository database. The dashboard ID for an individual dashboard is displayed on its Dashboard Features dialog in the iDashboards application. Dashboard IDs for multiple dashboards may also be read directly from the repository database, using an ISQL or reporting tool. The following query will produce a list of dashboards, including their dashboard ID numbers (the dash_id column), names and descriptions, as well as their associated categories:

```
SELECT d.dash_id, d.dashname, d.dashdescr, c.category_id,
c.category_descr

FROM fv_dashboard d, fv_dashcategory c

WHERE d.category_id = c.category_id

AND c.category_id <> 1
```

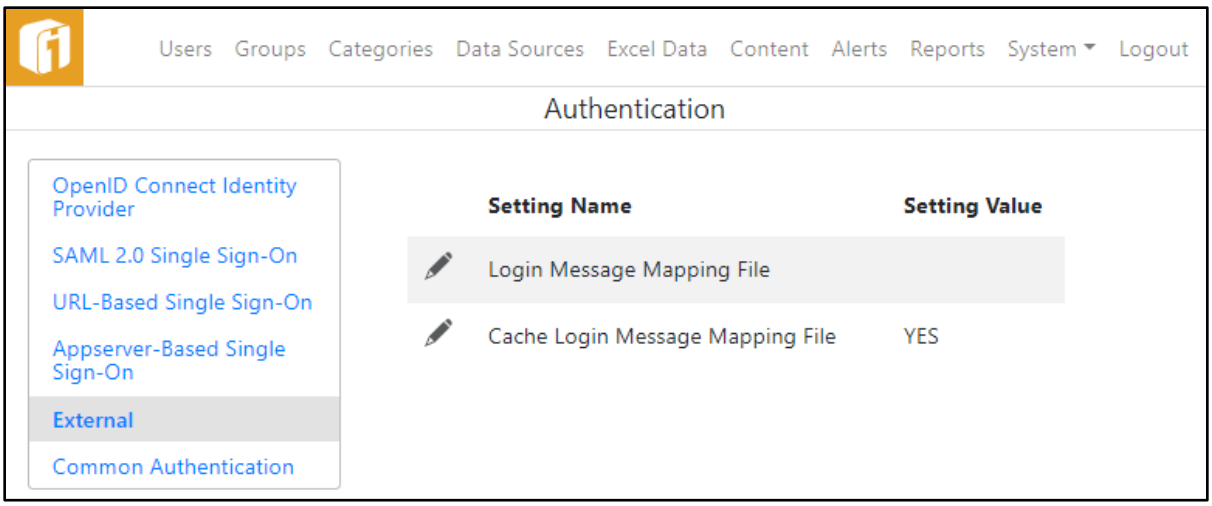
The condition "c.category_id <> 1" will exclude dashboards that are associated with users' Personal categories.

13.3.7 External

Allows the use of an external authentication module.

The setting "Login Message Mapping File" sets up the the path to an XML file which maps error messages returned by an external authentication module to error messages that are displayed to the user. If this setting is blank, iDashboards will look for a file called loginmessages.xml in the <IVIZGROUP HOME>\config directory. If no mapping file exists, then any error messages returned by an external authentication module will be displayed to the user upon a failed login.

For "Cache Login Message Mapping File", if set to YES, the Login Message Mapping File will be cached in memory, and re-read upon server restart. If it is set to NO, it will be re-read from disk on each failed login.

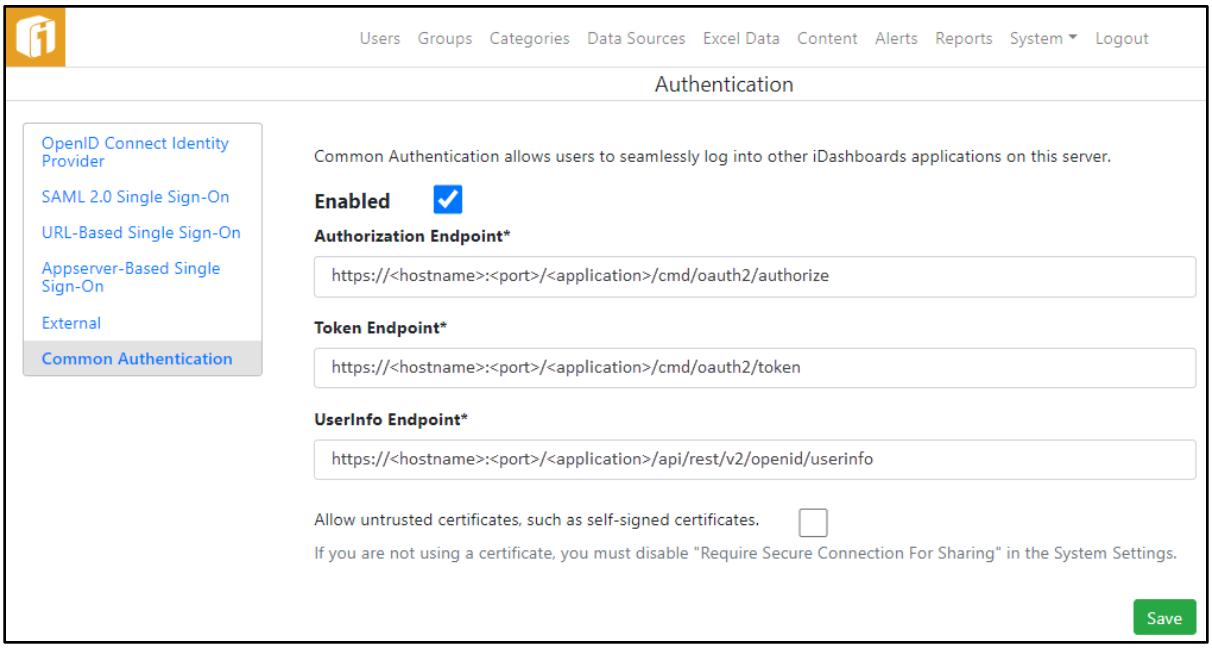


13.3.8 Common Authentication

This can be enabled when other iDashboards applications (ie Data Hub, Scorecards) are installed on the same server using the same iDashboard's repository database. It provides a single-application login to be used to access any of the installed applications from the iDashboard's Lobby.

To use this feature select the Enabled checkbox. The endpoint URLs are pre-configured, and should not need to be changed.

To allow untrusted certificates, such as self-signed certificates, select the checkbox for that. If you are not using a certificate, you must disable "Require Secure Connection For Sharing" in the System Settings.



13.4 Password Reset

13.4.1 Notification Email Settings

- **Password Reset Email Enabled:** If this setting is "No", then all email notifications, for password reset requests and server event notifications, will be disabled, and 'Forgot Password?' will not appear on login screen. If it is "Yes", then the settings in the SMTP Settings category must be properly configured to connect to the SMTP Service.
- **Password Reset Email "From" Address:** This setting must be a valid email address that will appear in the "From" header of notification email messages, for example "password-reset@mycompany.com". This setting is required if email notifications are enabled.
- **Password Reset Email "From" Name:** This optional setting is the name that will appear before the email address in the "From" header of notification email messages, for example "iDashboards Password Reset".
- **Password Reset Notification Subject:** This optional setting is a string that will be used to build the subject line for password reset emails.
- **Server Events Notification Threshold:** This setting determines what types of server events will generate emails to the addresses listed in the Server Events Notification List setting. The higher in the list a selection is, the fewer notification emails will be sent. Since a selection represents a threshold, each selection implicitly includes the ones above it.

  Selecting "Disabled" will turn off all email notification of server events.

  INFO-level events include the server starting up or being restarted after a pause. WARNING-level events include the server being shut down (in an orderly manner by the application server) or paused. ERROR-level events include any conditions that prevent the iDashboards Server from functioning properly.

- **Server Event Notification List:** This setting is a list of email addresses that will receive notifications of server events. Each email address should be on a separate line. Email addresses may be in plain format, for example:

  jsmith@company.com

  or in any RFC822-compliant format, for example:

  "Jane C. Smith" jsmith@company.com

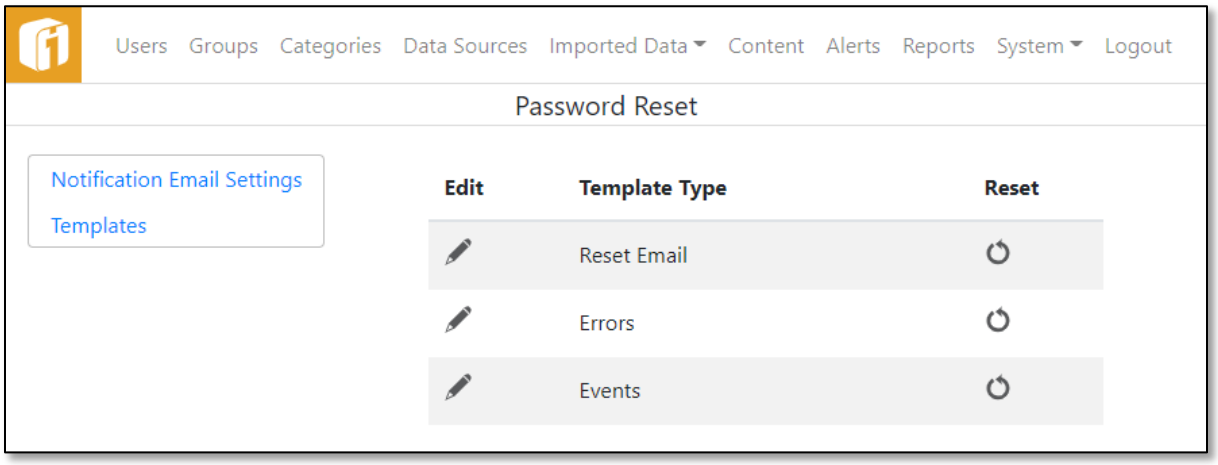
  The combined length for all email addresses, including end-of-line characters, must not exceed 500 characters.

- **Server Event Subject:** This optional setting is a string that will be used to build the subject line in server event notification emails.
- **Server Error Subject:** This optional setting is a string that will be used to build the subject line in server error notification emails.
- **Reset Request Expiration:** The amount of time (in minutes) that the password reset request will be valid for. Valid values are 0 through 9999.

13.4.2 Templates

This is an optional step that provides a great deal of control over the information included in the bodies of notification emails. Using templates, notification emails can be sent in both HTML format (including images) and plain text. If left untouched, notifications will be sent as plain text and include only a minimal amount of default information.

Templates are managed through the Templates screen. To access the Templates screen, select Templates from Password Reset.

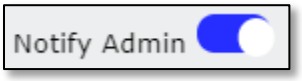


There are 3 Password Reset Template Types:

Type	Format	Usage
Reset Email	HTML/Text	Password Reset by Email
Errors	HTML/Text	Password Reset Error Notifications by Email.
Events	HTML/Text	Password Reset Event Notifications by Email.

To modify a Template, click its Edit icon ( ). The HTML panel provides a rich text editor for formatting and laying out an HTML emails, and the Text panel has a simple plain text box for plain text messages.

Each Template allows control over Administration notification by using the 'Notify Admin' switch.



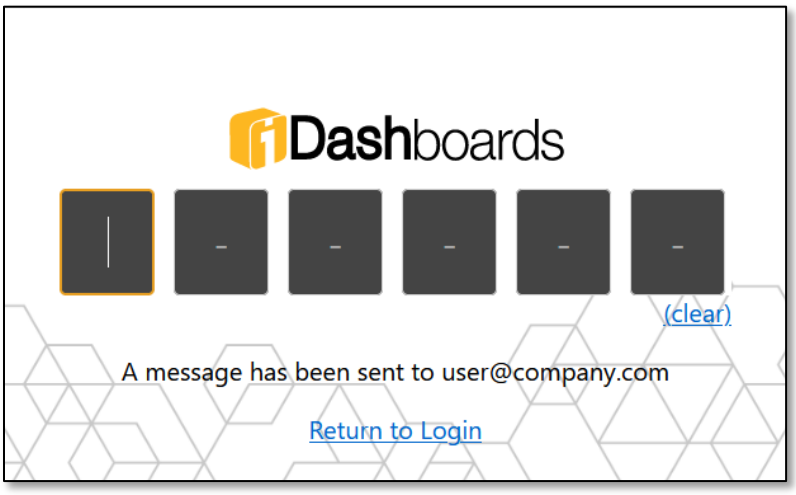
The first time a Template is edited the system creates sample HTML and Text notifications, with examples of using the available alert macros. Macros can found and selected by using the Macros button. The Help button will also provide details about the template macros.

After the first save of a template, the template files are created in an iDashboards' application directory, identified by the ${TEMPLATE_DIRECTORY} macro.

Using a Template's Reset icon () will regenerate the system created sample HTML and Text notifications for it, and also create the template files in ${TEMPLATE_DIRECTORY}.

13.5 Multi-Factor Authentication

Built into the core of iDashboards, is a framework to handle user authentication through a password. Additionally, a system administrator can configure Multi-Factor Authentication (MFA), with a second step requiring a six-digit access code that is sent to the user's email address.



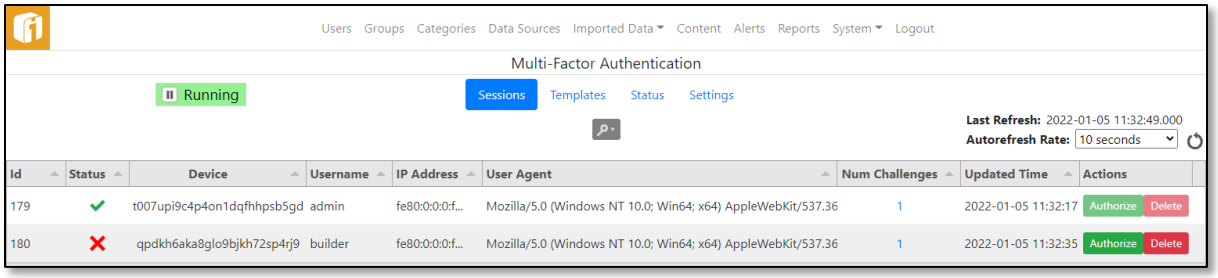
13.5.1 Email Configuration Roadmap

For the MFA Server to send email notifications, it must first be properly configured. The overall steps to accomplish this are:

- **Configure the SMTP (Simple Mail Transfer Protocol) Settings** – Notification emails are sent through an external SMTP service, such as UNIX Sendmail or Microsoft Exchange Server. It must be configured with enough information to connect to, and if necessary, authenticate itself to the SMTP service. (See Section 13.2.4 "SMTP Settings")
- **System Security Setting** – "Require User Email" needs to be set to TRUE. See Section 13.2.3 , "Security Settings"
    - The user's email address is maintained by an administrator through the "Users" page (See Chapter 7 "Managing Users"), or by the user through their individual "User Settings".
- **Configure the MFA Notification Email Settings** –
    - Email Sending Enabled – TRUE
    - Notification From Address – The email address used in the "from" header of outgoing emails.

13.5.2 Sessions

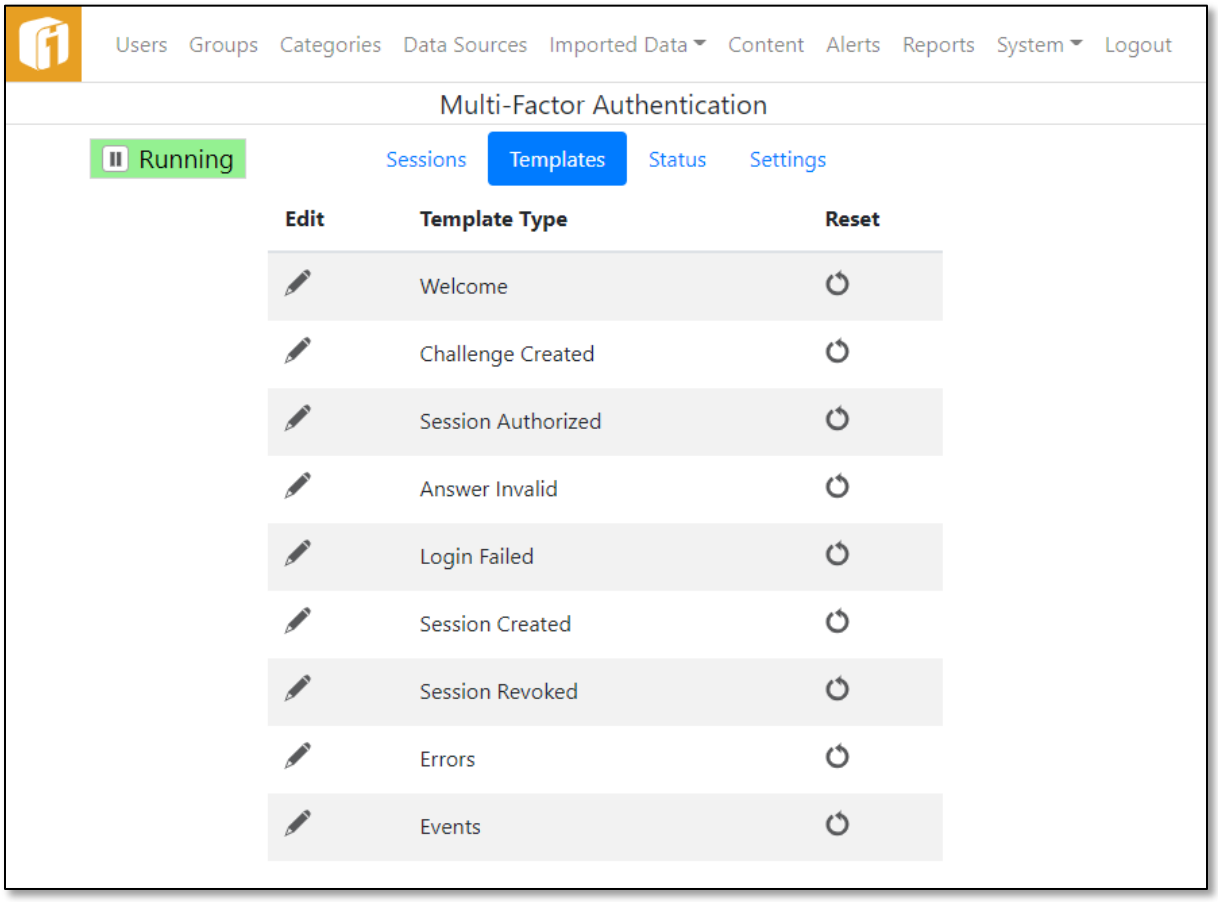
Sessions screen provides information and control of MFA login attempts. Status will show if it is Authorized or Pending. If "Pending" the number of challenges created will show a list of challenges for that specific session, with the ability to resend the authorization code email, or completely delete the challenge. Other controls facilitate manually authorizing or completely deleting the session. To access the Sessions screen, select Sessions from Multi.



13.5.3 Templates

This is an optional step that provides a great deal of control over the information included in the bodies of MFA emails messages. Using templates, MFA emails can be sent in both HTML format (including images) and plain text. If left untouched, notifications will be sent as plain text and include only a minimal amount of default information.

Templates are managed through the Templates screen. To access the Templates screen, select Templates from Multi-Factor Authentication.



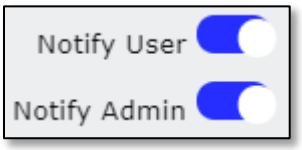
There are 9 Multi-Factor Authentication Template Types:

Type	Usage
Welcome	New User Welcome Message
Challenge Created	The Authorization Code
Session Authorized	A new device has been authorized
Answer Invalid	An invalid authorization code has been provided
Login Failed	An incorrect password was provided
Session Created	A login has been detected from an unknown device
Session Revoked	A Device has been revoked
Errors	MFA Server Error Notifications
Events	MFA Server Event Notifications

To modify a Template, click its Edit icon ( ). The HTML panel provides a rich text editor for formatting and laying out an HTML emails, and the Text panel has a simple plain text box for plain text messages.

The first time a Template is edited the system creates sample HTML and Text notifications, with examples of using the available alert macros. Macros can found and selected by using the Macros button. The Help button will also provide details about the template macros.

Templates also provide control over Administration and User notification by using the 'Notify Admin' and 'Notify User' switches.

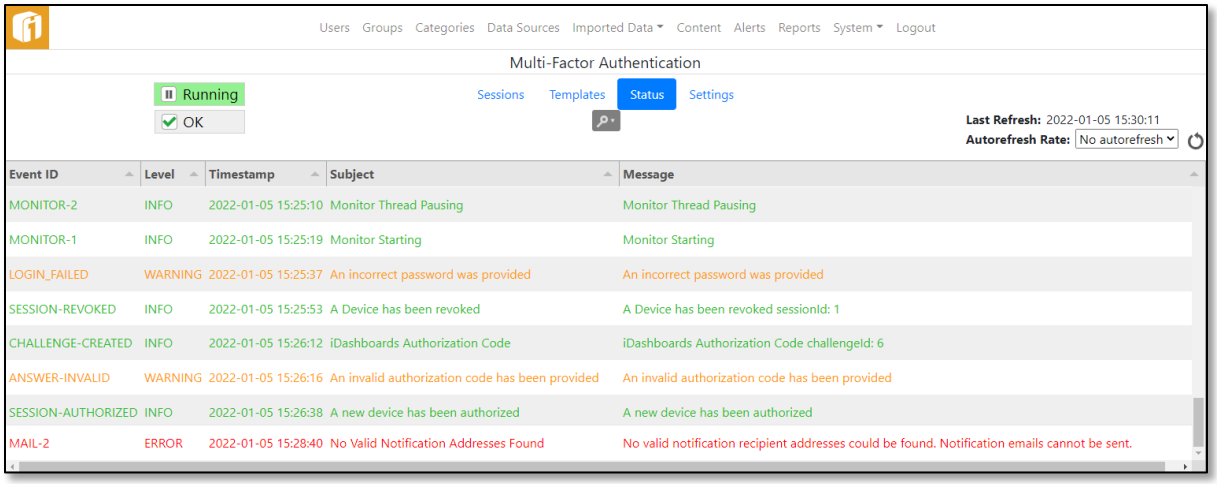


After the first save of a template, the template files are created in an iDashboards' application directory, identified by the ${TEMPLATE_DIRECTORY} macro.

Using a Template's Reset icon ( ) will regenerate the system created sample HTML and Text notifications for it, and also create the template files in ${TEMPLATE_DIRECTORY}.

13.5.4 Status

Within the Multi-Factor Authentication, an error or event can occur at any time and go unnoticed, and as a result, sessions might fail to generate when they should, or authorizations may fail. Multi-Factor Authentication provides a Status screen through which its inner workings can be observed. To access the screen, click "Status" from the Multi-Factor Authentication menu.



13.5.4.1 Pausing and Restarting Multi-Factor Authentication Server

At any given moment, the Multi-Factor Authentication Server will be in one of two possible states:

- **Running** – In this state, the Multi-Factor Authentication Server is performing all of its normal activities, such as session authorization, sending emails, etc.
- **Paused** – In this state, the Multi-Factor Authentication Server does not perform activities such as session authorization and sending emails; however, the Multi-Factor Authentication Server console is still fully functional.

In its default configuration, the Multi-Factor Authentication Server enters the paused state when it is started. When it is in the paused state, the "State" button will show "Paused". A paused server can be started by clicking the button, which will relabel it "Running". It can be placed back into the paused state by clicking the button.

13.5.4.2 Understanding Server Events

The most prominent feature of the Multi-Factor Authentication Server Status screen is the list of server events. A server event can be any type of noteworthy occurrence, such as the server being paused, an authorization code created, login detected. The event list can be filtered, using the search icon (), to only display events of certain, selected levels. This is accomplished by checking or unchecking the checkboxes for the different event levels.

A server event has the following attributes:

- **Event ID**
  Each server event is assigned a code referred to as the "event ID", which identifies the type of event that it is. And event ID consists of an event category, such as "MONITOR", and a number, separated by a hyphen.
  The event category is used to identify approximately where in the system the event occurred. For example, the MONITOR category is for events that occur on the monitor thread, which is the main thread that runs continually inside the server, checking alerts and performing other tasks.

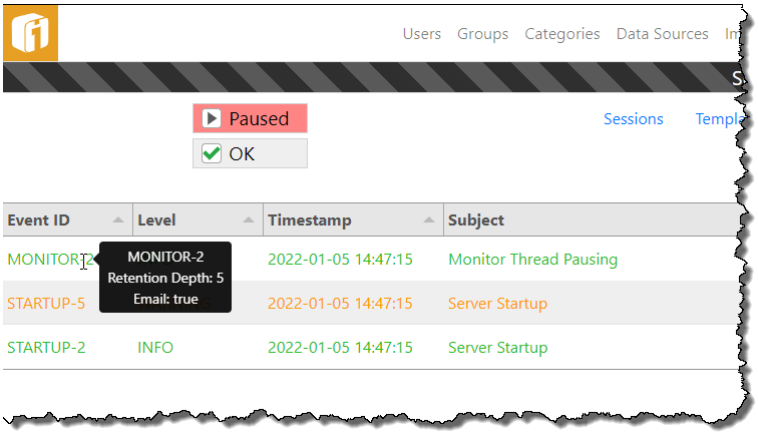
The number portion of the event ID uniquely identifies the type of event within an event category. For example, "MONITOR-2" is the event ID used to indicate that the monitory is being paused.

- **Level**
  Each server event has one of the following three levels:
  - **INFO** – This level is used for routine events. INFO-level events are displayed in green text in the event list.
  - **WARNING** – This level is for events that occur during normal operation, but should be noted by a server administrator. WARNING-level events are displayed in the yellow text in the event list.
  - **ERROR** – This level is used for abnormal, unexpected events such as a database error that occurs during alert generation. ERROR-level events are displayed in red text in the event list.
- **Timestamp**
  The event timestamp is the date and time at which the event occurred.
- **Subject**
  The event subject is a short phrase describing the event.
- **Message**
  The event message is a short sentence that contains information about the event.

13.5.4.3 Event Retention

During normal operation, the Multi-Factor Authentication Server is frequently recording new events in the event list. Because of this, one would expect that over time, the event list would grow extremely large, yet it does not. This is because only a certain number of events with a given event ID are retained in the event list. This number is referred to as the "retention depth" for that event ID. When the number of events with a particular event ID exceeds the retention depth for that ID, the oldest ones are removed from the list and discarded, keeping the entire event list at a manageable size.

The retention depth for an event ID is normally not of concern to the Multi-Factor Authentication Server administrator. It can be viewed, however, by holding the mouse cursor over the event ID in the event's list. This will produce a tool tip, similar to the one shown below, displaying the retention depth for the event ID.



13.5.4.4 Qualified Event Retention

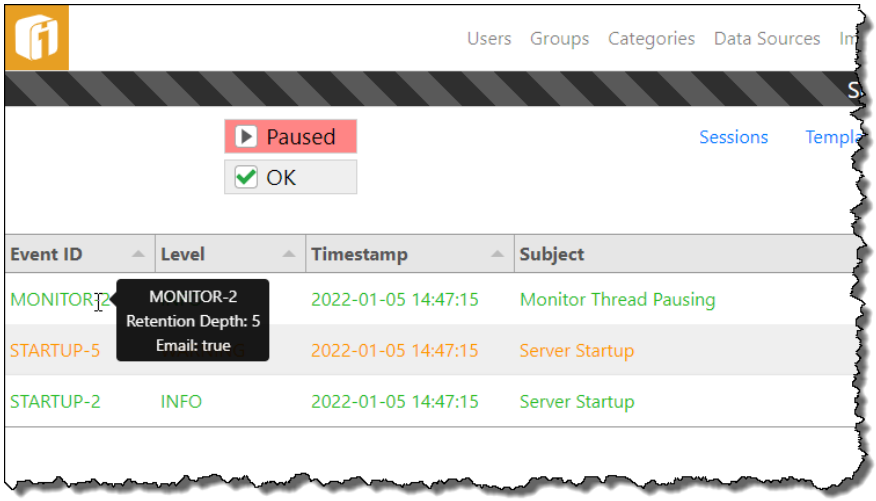
For some error events, the retention depth is not applied to the event ID alone, but rather to the event ID combined with some hidden qualifying information. This keeps important events from being pushed out of the event list before they can be viewed by an administrator.

13.5.4.5 Email Events

Certain event types are designated as "email events". When an email event occurs, a notification email will be sent to the designated Multi-Factor Authentication Server administrators, provided that:

- The Multi-Factor Authentication Server is properly configured to send event notification emails.
- The level of the event (INFO, WARNING, or ERROR) is at or above the configured threshold at which the event notification emails are sent.

To determine whether or not an event in the list is an email event, hold the mouse cursor over its event ID until the tool tip appears. It will include the line "Email: true" for email events, and "Email: false" for non-email events.



13.5.5 Settings

- **Multi-Factor Authentication Enabled**: This setting determines whether the Authorization Multi-Factor Authentication features are enabled in the user interface. Default is FALSE.

- **Email Sending Enabled**: Does this module send email? Default is FALSE.

- **Server Startup State:** This setting determines the initial state of the server upon startup. The two possible values are:
    - Running: The Authorization Monitor Thread will be started, and the server will check for expired objects according to its schedule. Default is Running.

  - Paused: The Authorization Monitor Thread will be in the paused state when the server starts up. It will need to be started manually on the Multi-Factor Authentication Admin page.

- **Force Authorization for All Users**: If enabled, all users that are not explicitly skipped will be required to authenticate all login attempts from unknown devices. Default is FALSE.

- **Skip Authorization List:** This setting contains a list of usernames (separated with a new line) that will never be required to authorize their device.

- **Force Authorization List**: This setting contains a list of usernames (separated with a new line) that will always be required to authorize their device. (The user will not be able to opt-out)

- **Device Prune Age (minutes)**: The amount of time (in minutes) that device records without a linked session are kept. Default is 0.

- **Challenge Prune Age (minutes)**: The amount of time (in minutes) that unconsumed authorization challenges are kept. Default is 30.

- **Session Prune Age (minutes)**: The amount of time (in minutes) that authorized user sessions are kept. Default is 0.

- **Unused Session Prune Age (minutes)**: The amount of time (in minutes) that unused user sessions are kept. Default is 30.

- **Check Interval**: The amount of time (in minutes) between authorization cleanup runs. Default is 5.

- **Notification From Address**: This setting must be a valid email address that will appear in the "From" header of notification email messages, for example "mfa@mycompany.com". This setting is required if email notifications are enabled.

- **Notification From Name:** This optional setting is the name that will appear before the email address in the "From" header of notification email messages, for example "iDashboards Multi-Factor Authentication". Default is "Multi-Factor Authentication".

- **Subject Notification**: This optional setting is a string that will be used to build the subject line in server event notification emails. Default is "Multi-Factor Authentication Notification".

- **Notification Threshold:** This setting determines what types of server events will generate emails to the addresses listed in the Server Events Notification List setting.

The higher in the list a selection is, the fewer notification emails will be sent. Since a selection represents a threshold, each selection implicitly includes the ones above it. Selecting "Disabled" will turn off all email notification of server events.

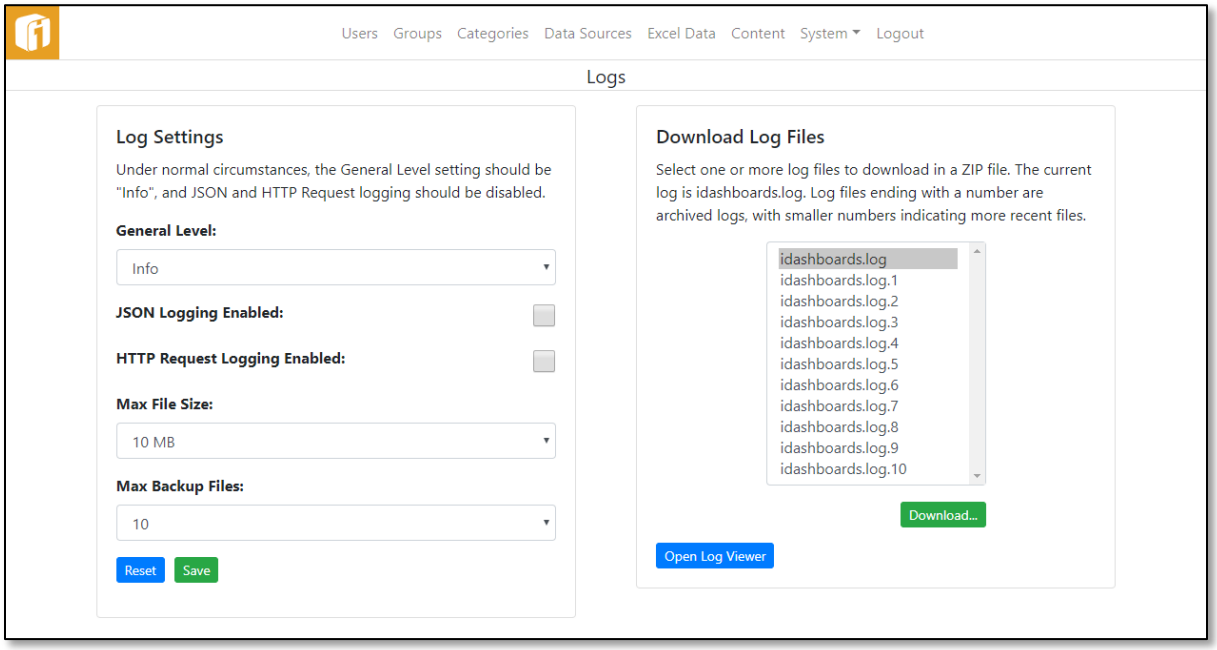
- INFO-level events include the server starting up or being restarted after a pause. Default is INFO.
- WARNING-level events include the server being shut down (in an orderly manner by the application server) or paused.
- ERROR-level events include any conditions that prevent the iDashboards Multi-Factor Authentication Server from functioning properly.

13.6 System Logs

13.6.1 Log Settings

Log settings are managed through the Log Settings screen. The Log Settings screen can be accessed by selecting menu item "System > System Logs".

When the iDashboards server is started, the initial log settings are read from the ivizgroup.properties file, and if they are not present, the defaults are used. Once the server has been started, the settings can be changed through the Log Settings screen, however, changes made will not persist beyond a server restart. Under normal operating circumstances, the settings shown below should be used.



Changes made to log settings are not applied until the "Save" button has been selected. The available settings are:

General Level — This setting determines the types of log messages that will be written to the log file. Each level can be thought of as a threshold, with Debug being the lowest and Error the highest. When a level is selected, all messages categorized at that level and above will be written to the log. The available levels are as follows:

- **Debug** — This is the most verbose setting and could impact system performance on a busy server. Debug log messages are generally only useful to an iDashboards support representative, so this level should only be used when troubleshooting.
- **Info** — This is a far less verbose level than Debug, which writes information about the operating environment to the log when the iDashboards server is started. **It is the recommended level for normal operations.**

- **Warn** — In addition to error messages, this level will write warning messages about server events that are noteworthy but not critical.
- **Error** — This is the least verbose log level. It will only write messages to the log when a critical error occurs.

JSON Logging Enabled — When this checkbox is checked, JSON data that passes between the iDashboards client and the server is written to the log file. It causes very verbose output which is only useful to an iDashboards support representative, so it should remain unchecked except for troubleshooting purposes.

HTTP Request Logging Enabled — When this checkbox is checked, information about the HTTP requests that are sent to the iDashboards server are logged. As is the case with JSON logging, it causes very verbose output which is only useful to an iDashboards support representative, so it should remain unchecked except for troubleshooting purposes.

Max File Size — This is the maximum size to which a log file will be allowed to grow before it is overwritten by a new one or archived.

Max Backup Files — This setting indicates the maximum number of archived log files that will be kept. When an active log file, named "idashboards.log" grows to its maximum allowed size, it will be renamed with a numeric suffix "idashboards.log.1" and a new idashboards.log file will be started. If there is already a file named idashboards.log.1 in the logs directory, it will be renamed idashboards.log.2, and so on, up to the maximum number of archived log files. When the maximum number has been reached, the oldest archived log file will be discarded.

Log File Name	Relative Date	Relative Age	Status
idashboards.log	Today	Newest	Active
idashboards.log.1	Last Week		Archived
idashboards.log.2	Last Month	Oldest	Archived

13.6.2 Downloading Log Files

The active log file (idashboards.log) and any existing archived log files (idashboards.log.1, idashboards.log.2, etc.) can be downloaded through the Log Settings screen. To do so, select the desired files from the list at the right of the screen and select the "Download…" button. The selected files will be bundled into a ZIP file by the server and downloaded.

13.6.3 Sending Log Files to iDashboards Technical Support

When working with iDashboards technical support to troubleshoot problems with the iDashboards server, it is useful to provide the iDashboards log file(s) to the support representative. Problems can be diagnosed and corrected more expeditiously if these steps are followed prior to contacting iDashboards technical support:

1. Set the General Level to Debug, and enable JSON logging and HTTP Request Logging.
2. Recreate the error condition through the iDashboards application.
3. Download the idashboards.log file and the idashboards.log.1 file if it exists, as described in Section 13.4.1, "Downloading Log Files".
4. Email the ZIP file containing the log file(s), along with a description of the problem (and steps to recreate it if possible) to support@idashboards.com.

13.6.4 Log Configuration

At runtime, iDashboards will log system errors and other events in a log file. The name of the log file is idashboards.log, and it will be created in the <IVIZGROUP HOME>\logs directory. Certain parameters can be set in the ivizgroup.properties file to determine the maximum size a log file will be allowed to grow to, the number of backups that will be kept, and the verbosity of the logging output.

Note: These settings can be changed while the server is running through the System Logs screen, described in Chapter 13, "System Configuration", however such changes will not persist across a server restart.

log.directory – This property can be used to indicate a directory other than <IVIZGROUP HOME>\logs where log files should be written. It must exist and be writable by the iDashboards application server process. Forward slashes (/) should be used instead of backslashes (\) as a path separator.

log.maxFileSize – This property indicates the maximum size, in bytes, that a log file will grow to before it is "rolled over", that is, renamed with a ".1" extension so that a new idashboards.log file can be created. This property must be an integer from 0 to 9,223,372,036,854,775,808. (Do not include commas.) The suffixes "KB", "MB", or "GB" can be appended to indicate the value is kilobytes, megabytes or gigabytes, respectively. If no value is given, the default used is "10MB".

log.maxBackupIndex – When logs are rolled over, the current idashboards.log file is renamed to idashboards.log.1, an existing log file with a ".1" extension is renamed with a ".2" extension, one with a ".2" extension is renamed with a ".3" extension, and so on up to the value of the log.maxBackupIndex property. If a log file already has an extension equal to log.maxBackupIndex, it is discarded when the log files are rolled over. If log.maxBackupIndex is zero, there will be no backup files, and the log will be truncated when its size grows to the maximum size.

log.level – This value must be one of the following: ERROR, WARN, INFO or DEBUG. The default is INFO. DEBUG will produce the most verbose output, and ERROR will produce the least. Normally, DEBUG should only be used when troubleshooting.

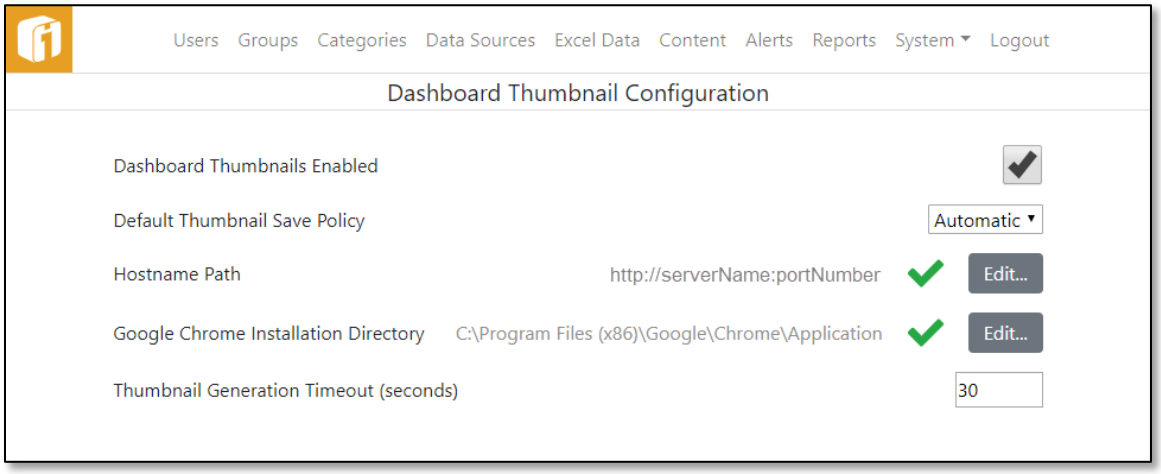
13.7 Dashboard Thumbnail Configuration

Dashboard Thumbnails are an optional feature of generating dashboard images to use, along with the dashboard name, as means to quickly identify it.

IMPORTANT!

iDashboards used the Google Chrome browser, in headless mode, to capture and create the dashboard images to be used as thumbnails. Chrome will only be used by iDashboards in headless browser mode.

Chrome does not auto-update while using headless mode. To update Chrome, one must manually launch the browser to begin the update process.



13.7.1 Dashboard Thumbnails Enabled

If TRUE, thumbnail images can be saved for dashboards that will be displayed to the user in the user application. If false, all dashboard thumbnail functionality will be disabled. The default value is TRUE.

13.7.2 Dashboard Thumbnail Save Policy

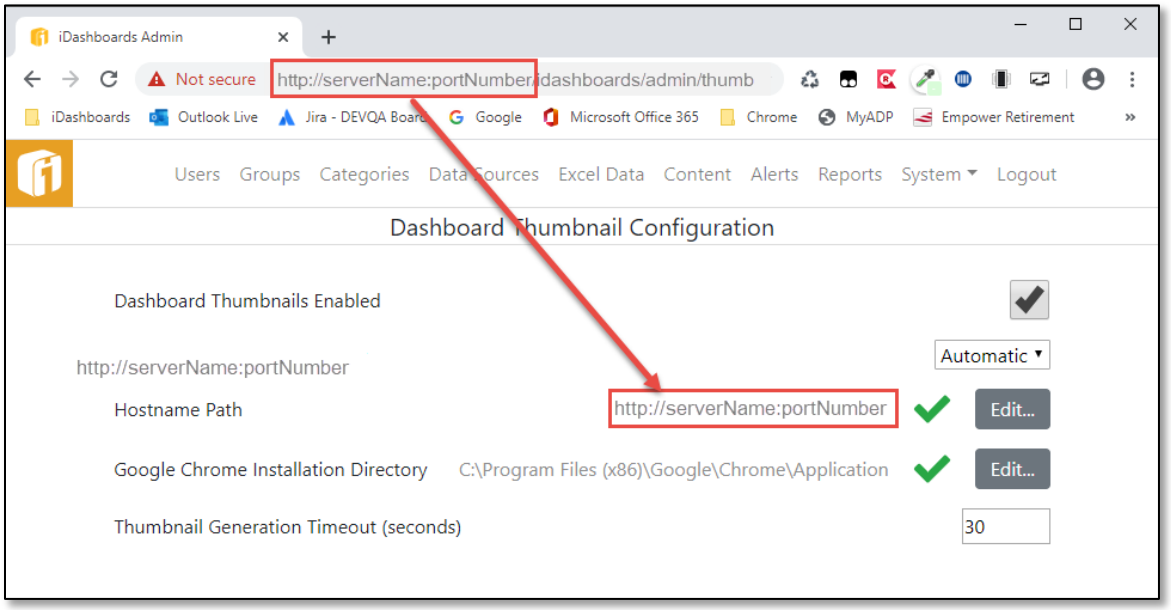
This setting determines when a dashboard thumbnail will be saved. The default value is Automatic. The available choices are:

- **Automatic**: A dashboard thumbnail will automatically be saved each time the dashboard is saved.
- **Prompt**: The user will be prompted when they save a dashboard to indicate if they would like to save a dashboard thumbnail.
- **Never**: The dashboard thumbnail will never be saved. These dashboards will be represented by a generic thumbnail.

13.7.3 Hostname Path

This is the URL that the iDashboards Application uses, in conjunction with the headless Chrome browser, to open a snapshot of the dashboard that is used to create the thumbnail image.

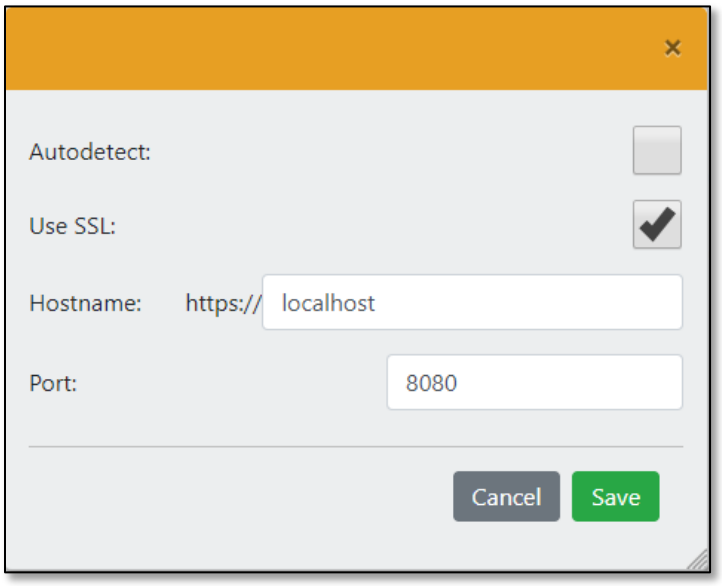
By default the value is automatically detected, and in most cases this will be the same Host URL used by the iDashboards application.



However, depending on the network settings, changes to this URL may be needed. To make changes select the "Edit" button. Unselect "Autodectect". If needed select "Use SSL" for an "https" connection. Provide the Hostname and Port number, and select "Save".

Other Hostnames possibilities are:

- localhost
- serverName (without the .companyName.com)
- IP Address (i.e. 123.456.789.10)



13.7.4 Google Chrome Installation Directory

This is the path to the Google Chrome application, used in headless mode, to capture and create the dashboard images to be used as thumbnails.

By default the value is automatically detected, and in most cases will work. But if needed the location of the Google Chrome application can be modified. To make changes select the "Edit" button. Unselect "Autodectect", and in the Chrome Installation Directory provide the full path and select "Save".

13.7.5 Thumbnail Generation Timeout (seconds)

This is the maximum amount of time that will be allowed for a thumbnail to be generated. Allowable values are from 5 to 300 seconds, with an initial default of 30 seconds.

Dashboards that normally take a little time to open and fully display, will also take a little more time when saving the dashboard and creating the thumbnail image. Dashboards that link to remote servers may also take a longer time to create the thumbnail. Under these scenarios, it may be necessary to adjust this timeout value.

13.8 Languages (Localization)

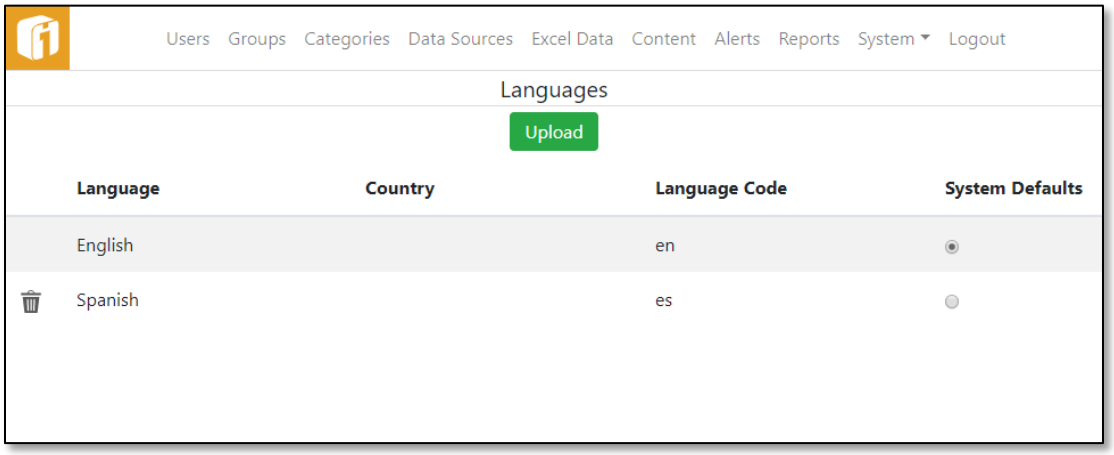
The default language for iDashboards is US English. iDashboards supports non-English languages through the use of Language Packs. Language packs need to be uploaded to iDashboards under the menu item "System > Languages".

Note: Contact iDashboards for availability of Language Packs.

Language packs affect only the text that is built in to the iDashboards application. For example, the language packs affect the text in the menu items, the right-click menu options, and the windows related to dashboard and chart creation. They also affect the text that administers the users, groups, categories, data sources and settings. Language packs do not affect any textual data drawn from a Data Source. Data Source texts are not translated by Language packs. Also, category names are displayed as created in the Administrator application. The iDashboards manuals are written in US English.

13.8.1 Installing Language Packs

Before a language pack can be used, it must be installed on the iDashboards server. This is done through the Languages screen of the Administrator application. To access the Languages screen, select menu item "System > Languages". The main languages screen will appear. This screen displays a list of the installed language packs and an associated country and country code. The default system language is identified by a selected option button.



To install a language pack, perform the following steps:

1. A language pack consists of a zipped file with the two letter abbreviation of the language as the file name and .zip as the file extension. Select the Upload button, then browse to and select the file on the disk.
2. The Administrator application will upload the file to the server, check them for consistency and either accept them or reject them. If accepted the list will be updated with the new language. Otherwise a status message will be displayed.

13.8.2 System Defaults

The iDashboards application have a language set as a default language. This is to ensure that all users utilize the same language but can personalize their own language as necessary. By default the language is US English. This can be changed by selecting the System Defaults option button for the language.

If there is no language pack then iDashboards will revert back to US English.

13.8.3 Deleting a Language Pack

A language pack may be deleted from the repository. The language pack may be deleted by selecting on its Delete icon ().

If a language pack is set by a user in their User Settings and the language pack is deleted, then the language will default back to the System Defaults language, if available.

13.9 Uploading Images to iDashboards

Dashboard frames can display external content, such as images, from outside of iDashboards. This feature allows dashboard builders the ability to add logos, graphics, and other external graphics to a dashboard.

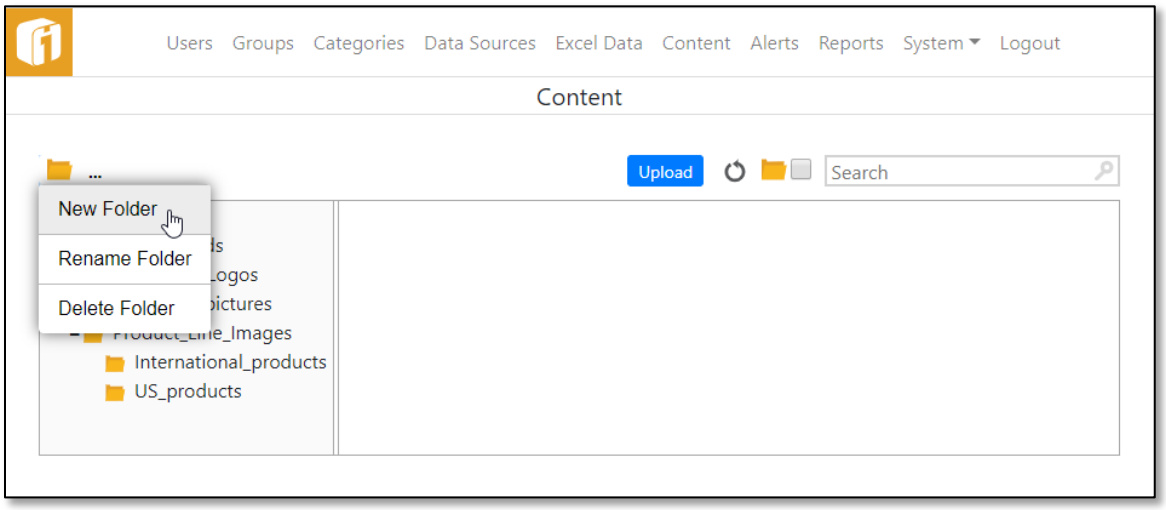
The iDashboards Administrator application provides a way for administrators to upload external content to the server hosting iDashboards, so it will be available to the iDashboards application. This is performed by selecting the "CONTENT" tile from the Administrator home screen, or "Content" on the menu bar.

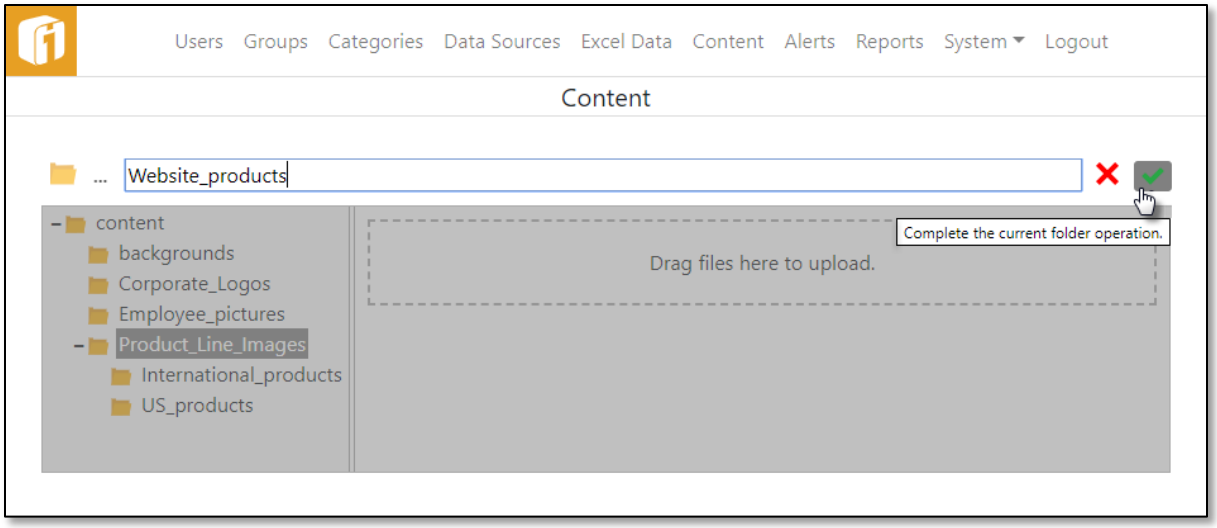
Not all file types can be utilized in iDashboards. The file types that can be used are:

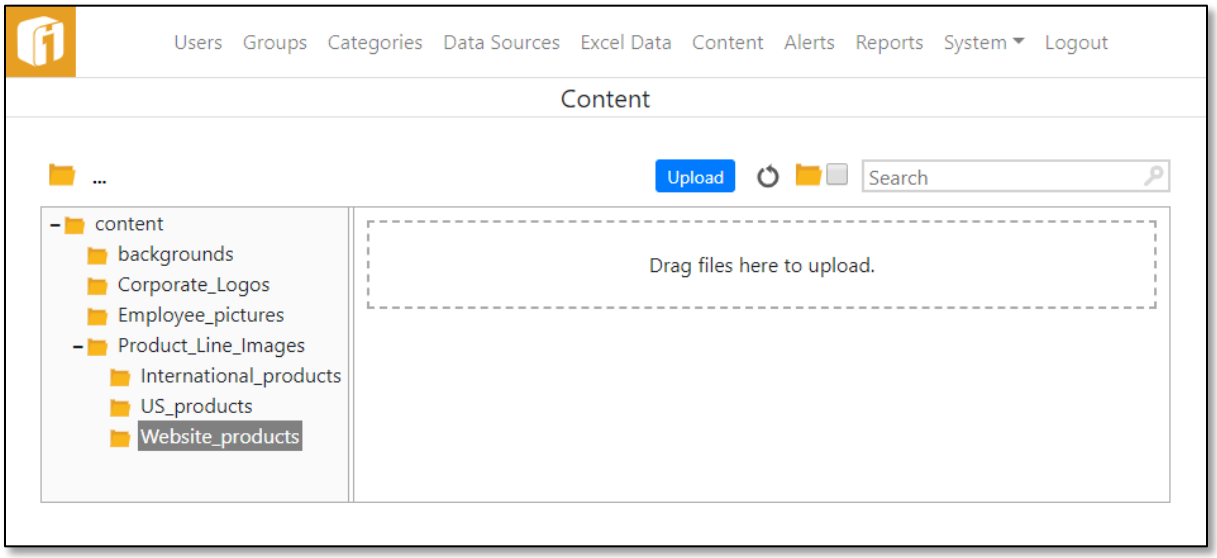
- .gif (non-animated only)
- .jpeg
- .jpg
- .png
- .xml (specifically designed for use with the Image Plot Chart)

13.9.1 Content Folder Structure

Before images are uploaded, consider file management through the use of multiple folder structures. To create a sub-folder, first select the "parent" folder to contain the new folder. Then select the "Folder Control" button, and then "New Folder". Enter the folder name and select the "complete" button.

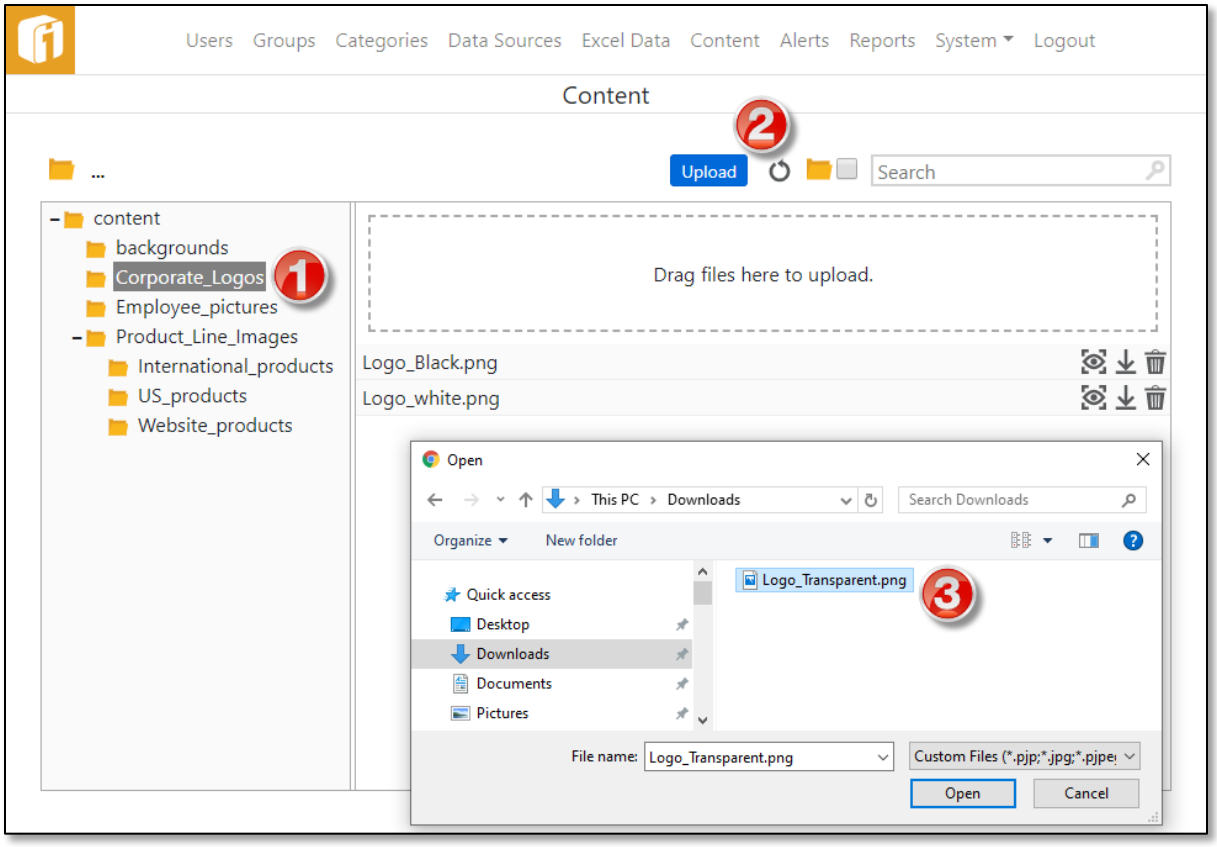




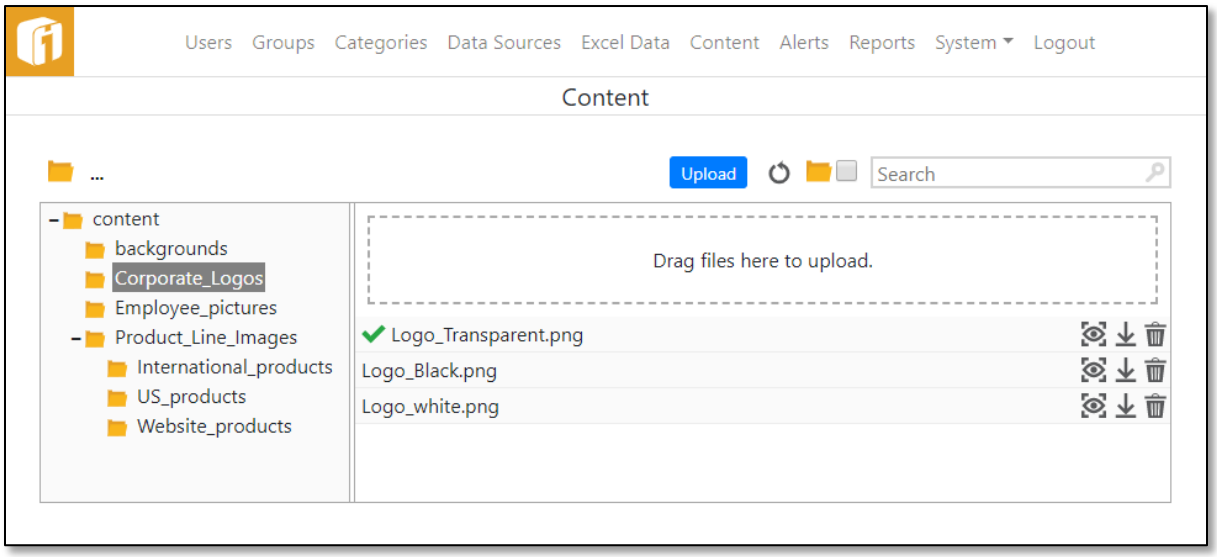


13.9.2 Uploading Content

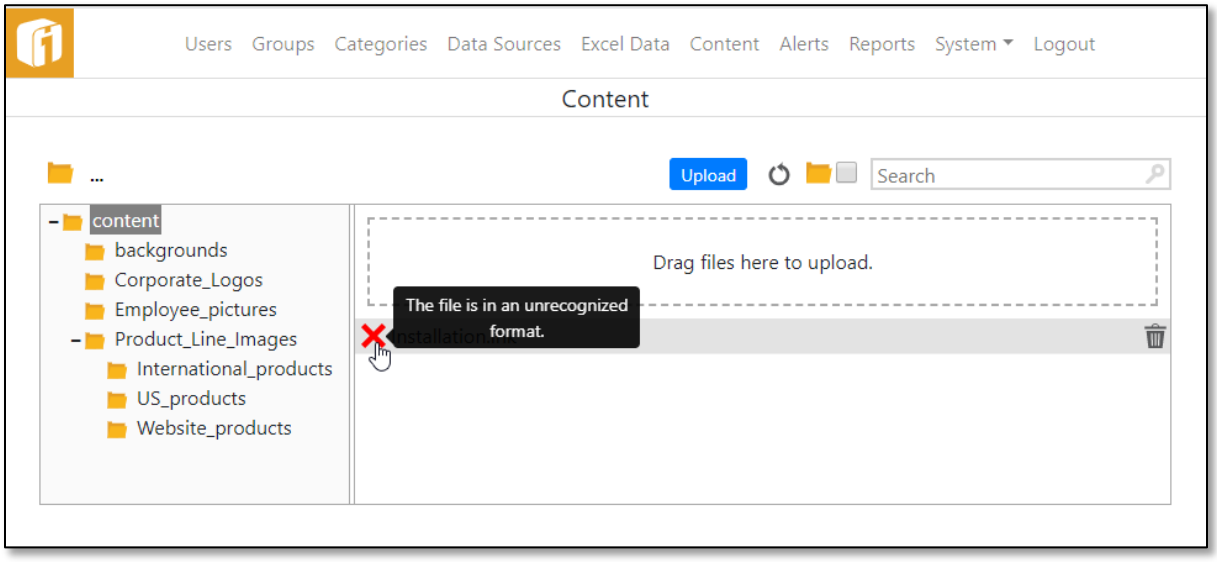
To upload a file, select the destination folder or sub-folder first (on the left), select the "Upload" button, then navigate to and select the file from your file system. Alternatively, drag the file from your file system to the "Drag files here to upload" area.



Uploaded files reside in the content folder in the ivizgroup home directory. If the file is properly uploaded to the <ivizgroup home>/content folder, it will show with a green checkmark.



If the file is not a recognized as a valid iDashboards file type then you will get the following error message, next to the file name:



To view the files that have been uploaded select the name of the file and then the View icon (). This will bring up a browser window with the image. You can also download the file by selecting the Download icon ().

13.9.3 Content Removal

To remove a file, select the name of the file and the Delete icon (). iDashboards will confirm before actual removal. Once removed, the file will no longer be accessible and is no longer located in the <ivizgroup home>/content folder. The file can be removed even if it is embedded in a dashboard. If a file gets removed and it is still in the frame of a dashboard then the dashboard will display a broken image icon.

13.10 Importing and Exporting

Items can be exported from an iDashboards repository (called the source repository) and imported into another iDashboards repository (called the target repository). Charts, dashboards, picklist, forms (if enabled), and knowledge base articles (if enabled), are exported to an iDashboards archive file (named with the .zip extension). This file can then be referenced during the import process for the target iDashboards repository.

Note: The Forms and Knowledge Base features are enabled within the iDashboards license.

Note: Charts, dashboards, picklists and forms (if enabled) that are stored in the "Personal" Category cannot be exported because the "Personal" Category represents a personal workspace rather than an enterprise-wide workspace.

Each chart, dashboard and picklist that will be exported and/or imported must be assigned a Global Identifier (GID). GIDs are enterprise-wide identifiers that are used to identify charts, dashboards and picklists across multiple installations of iDashboards. To fully support the import/export functionality, GIDs can also be assigned to categories, Data Sources and stored procedures (and may be required as explained later in this section).

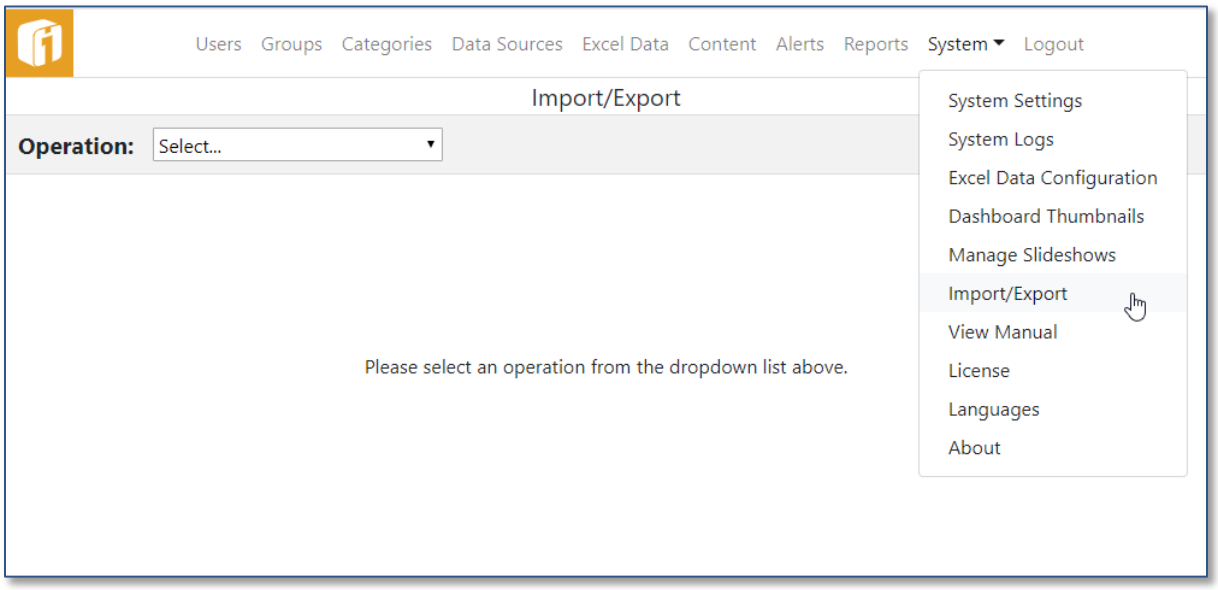
Note: The GIDs that are assigned to stored procedures are assigned to the stored procedure that is created within iDashboards, not the actual stored procedure that resides in the data source. The stored procedure that is created within iDashboards is, in essence, configuration information about the actual stored procedure that exists in the data source.

By default, GIDs ***are not*** assigned to charts, dashboards, picklists, categories, data sources stored procedures and forms (if enabled) when they are created. An iDashboards administrator must manage GIDs as described later in this section.

Note: GIDs are not the same as the local IDs that are automatically assigned to all charts, dashboards, etc.

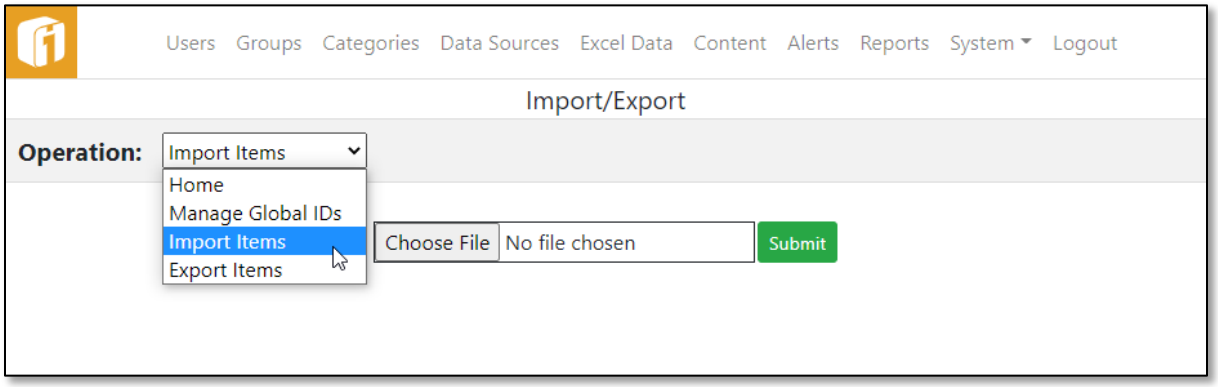
For Knowledge Base articles (if enabled), there is no way to assign a Global ID. Instead, every article has a permanent, globally unique, non-editable identifying code assigned to it when it is first created by a user and saved to the repository. This code serves the same purpose as a Global ID, which is to determine whether or not two articles in two different repositories are, for import/export purposes, effectively the same article. To distinguish this code from a normal iDashboards Global ID, we refer to it as a GUID, which is short for "Globally Unique ID."

The importing and exporting of charts, dashboards, picklist, forms (if enabled), and knowledge base articles (if enabled), is managed through the "Import/Export" option under the System menu of the Administrator Application.



The administrator has the option to select one of the following from the "Operation" drop-down list:

- Manage Global IDs
- Import Items
- Export Items



13.10.1 Criteria for Export

Charts, dashboards, picklists and form (if enabled) may be exported separately. Each has their own criteria for exportation.

For a chart to be eligible for export, the following conditions must be met:

1. The chart must have been saved under iDashboards version 6 or later.
2. The chart must be assigned a GID.
3. The chart's Data Source must be assigned a GID, unless the chart uses static data.
4. The Category the chart is saved in must be assigned a GID.
5. If the Data Source for the chart is a stored procedure, the stored procedure must be assigned a GID.
6. If the chart has a drilldown and that drilldown is to be maintained, the chart or dashboard the drilldown points to must adhere to the chart and dashboard export criteria outlined in this section.

Note: If a chart has a drilldown, and the chart or dashboard the drilldown points to, is not given a GID, when exported from the source repository and imported into the target repository, the drilldown link on the chart will be removed when the chart is imported into the target repository.

For a dashboard to be eligible for export, the following conditions must be met:

1. The dashboard must have been saved under iDashboards 6 or later.
2. The dashboard must be assigned a GID.
3. The Category the dashboard is saved in must be assigned a GID.
4. All charts and picklists within the dashboard must adhere to the entities export requirements (described above).

For a picklist to be eligible for export, the following conditions must be met:

1. The picklist must have been saved under iDashboards 6 or later.
2. The picklist must be assigned a GID.
3. The picklist's Data Source must be assigned a GID, unless the picklist uses static data.
4. The Category the picklist is saved in must be assigned a GID.
5. If the Data Source for the picklist is a stored procedure, the stored procedure must be assigned a GID.

Note: If the Forms feature is enabled, within the iDashboards license, exporting forms is also included.

For a form to be eligible for export, the following conditions must be met:

1. The form must have been saved under iDashboards 10.5 or later.
2. The form must be assigned a GID.
3. The form Data Source must be assigned a GID.
4. The Category the form is saved in must be assigned a GID.

Note: If the Knowledge Base feature is enabled, within the iDashboards license, exporting articles is also included.

Articles are always valid for export

13.10.2 Criteria for Import

Before importing takes place the data sources, and their Global ID's, must match between the two systems. The "IDB System Database" is the System Database which iDashboards uses to store metadata. By default the ID of this repository will be zero. However, if during the installation process the ID was purposely changed to another value, the ID will need to be changed within the second system before import. Other data sources must also have matching ID's before importing.

For a chart or picklist to be eligible for import, the following conditions must be met:

1. There must be a Data Source in the target repository with the same GID as the one associated with the chart or picklist that will be imported.
2. If a stored procedure is used to retrieve its data, the database the imported chart or picklist will point to must have a stored procedure identical to the stored procedure the imported entity pointed to in the source database.

Note: Stored Procedure definitions in iDashboards don't need to be created manually in the target repository. They will be imported automatically when the .zip file is imported, assuming the stored procedure was given a GID during the export process. The stored procedure definitions are imported along with the charts that use them.

For a dashboard to be eligible for import, the following conditions must be met:

1. The dashboard must have a GID.
2. The Category the dashboard is saved in must have a GID.
3. All charts and picklist within the dashboard must adhere to the entities import requirements (described above).

Note: If the Forms feature is enabled, within the iDashboards license, importing forms is also included.

For a form to be eligible for import, the following conditions must be met:

1. There must be a Data Source in the target repository with the same GID as the one associated with the form that will be imported.
2. The Category the form is saved in must have a GID.
3. Because forms can have permissions based on groups, and groups don't export, upon import, the "by group" forms will have zero permissions until they are defined.

Note: If the Knowledge Base feature is enabled, within the iDashboards license, exporting articles is also included.

Articles are always valid for import.

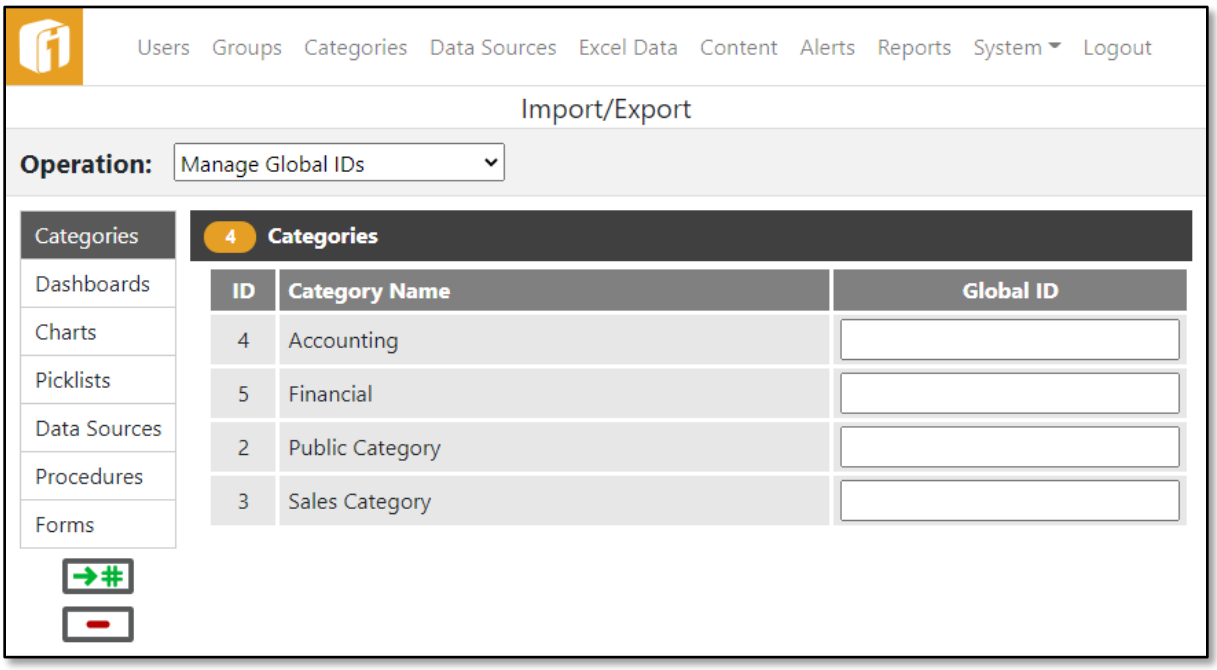
13.10.3 Additional Elements during Import/Export

Images and the Content folder structure need to match on both the Source and Target servers. Failure to keep the content structure and files the same could result in broken-image icons and missing graphics on a chart or dashboard.

- **Images can be used in these functions** – Chart background, Chart Reference via Chart (Image Plot, SVG Drawing, Gallery, Slideshow, Details, or Infographics), Dashboard background, or Dashboard Frame
- **XML files** – Image Plot charts might use .XML files

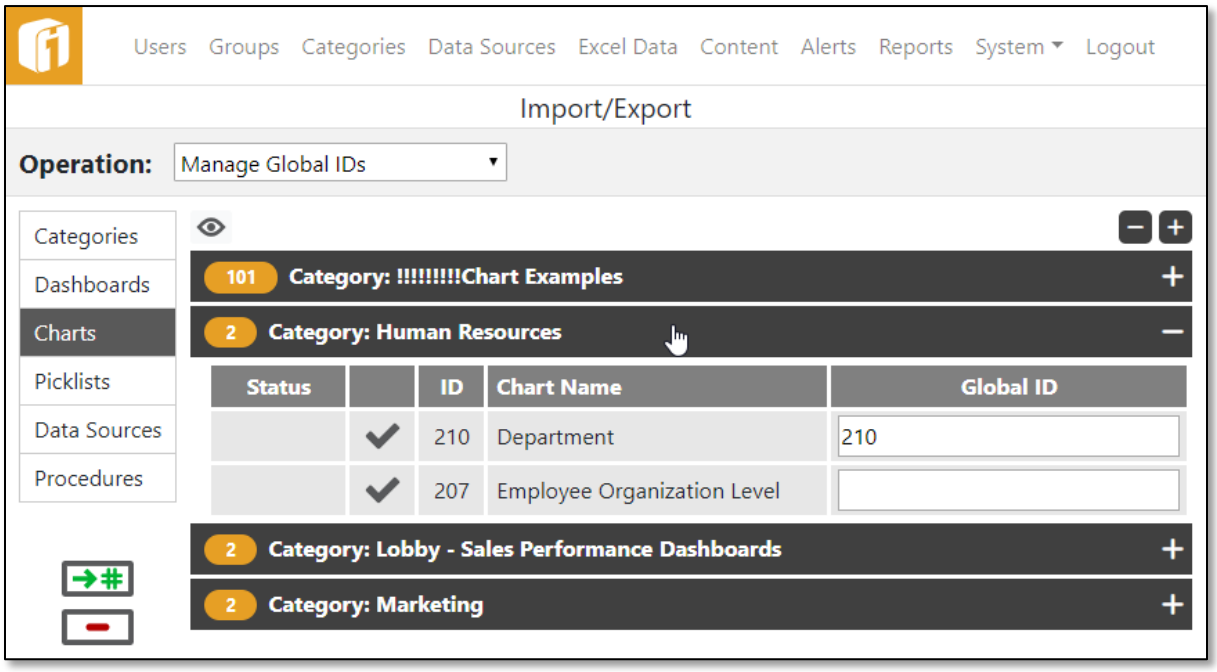
13.10.4 Managing Global Identifiers (GIDs)

When an administrator selects "Manage Global IDs" from the "Operation" dropdown box, they will be presented with a list of the six iDashboards entity type groups that can be assigned GIDs (Categories, Dashboards, Charts, Picklists, Data Sources and Procedures). GIDs are managed by selecting the group.



By default, no entities in iDashboards are assigned a GID when they are created. To change this, either select the → # button on the "Import/Export" screen or set the system parameter called "Generate Global IDs", under "Server Settings", on the "System Settings" screen, to True. When iDashboards is configured to generate GIDs automatically, every new entity that is created will simply get assigned the next numeric GID available. This is the reason GIDs are not assigned by default. For example, if two installations of iDashboards are configured to assign GIDs automatically, then the administrator has no control over the GIDs that are assigned to the various entities. Therefore, when the administrator chooses to import and export, they cannot be assured the GIDs will match up how they would like between the two installations. It is safer for the administrator to manually manage GIDs.

Selecting one of the groups, Categories, Dashboards, Charts, Picklists, Data Source, Procedures or Forms (if enabled), will display a list of entities of the indicated type that are available for GID management. The list of entities will be grouped if applicable. Data Sources and categories will not have any groupings. Procedures will be grouped by the Data Source they are associated with. Dashboards, charts and picklists will be grouped by their categories. To set the Global ID for those entities that are within a group you will need to click on their corresponding group to expand and view the entities. The expandable link will have a finger similar to a hyperlink when you hover over it. For the Dashboards, Charts Picklists and Forms (if enabled) groups, the icon will toggle the showing of empty categories.



If an entity has an associated Global ID it will be displayed in the "Global ID" column. To change or add a Global ID, enter the unique number in the "Global ID" column. The administrator must assign/update each GID individually.

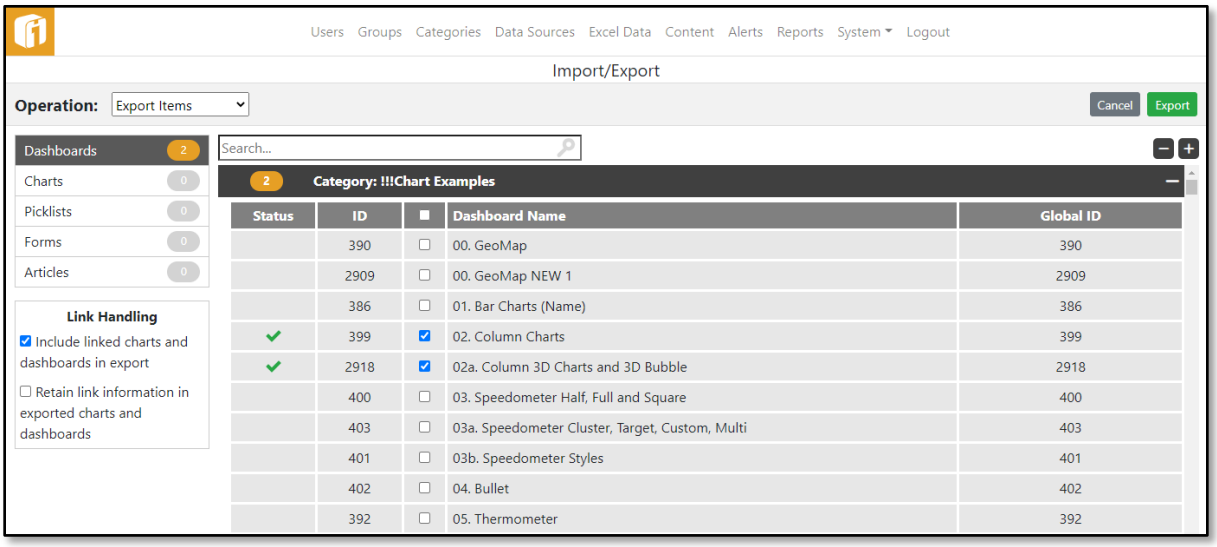
> *Note: GIDs are assumed to be unique within an entity type; in effect, the same GID cannot be assigned to two different charts but a chart and a dashboard can have the same GID. If there is a GID conflict within an entity type, an error message will appear indicating that GIDs must be unique.*

13.10.5 Exporting Items

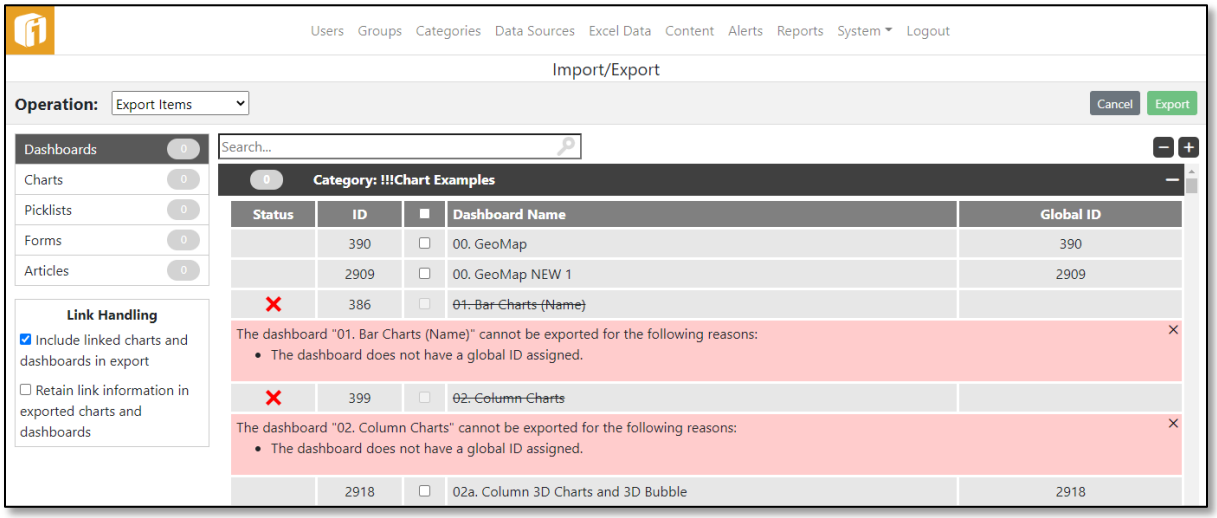
When an administrator selects "Export Items", from the "Operation" dropdown box, the export groups, (Dashboards, Charts, Picklists), will be available to select from. Selecting one of the groups will show a list of its available categories. To select the dashboards, charts, or picklists within the category you will need to check the box next to the corresponding item. An administrator will also have the option of selecting, or unselecting, all items in the category by checking the box at the top in the categories column header line.

Note: If the Forms feature is enabled, within the iDashboards license, exporting forms is also included.

Note: If the Knowledge Base feature is enabled, within the iDashboards license, exporting articles is also included.



Items that do not meet the criteria set forth in Section 13.8.1, "Criteria for Export" cannot be exported and are identified with a strikethrough on its name. If select a ✕ will show in the Status column. Clicking on the ✕ will bring up a dialog box that will describe why the entity cannot be exported.



Under "Link Handling" there are 2 options. Select either option, or neither option, but the system will not allow the selection of both.

1. Include linked charts and dashboards in export (default selected)

All of the charts and dashboards in a chain of drilldowns from a selected dashboard or chart will be included, provided that they meet the criteria set forth in Section 13.8.1, "Criteria for Export". A drilldown chain will be followed until a non-exportable target is encountered, at which point it will be followed no further.

2. Retain link information in exported charts and dashboards (default un-selected)

   All of the charts and dashboards in a chain of drilldowns from a selected dashboard or chart will NOT be included, but the info about their drilldown Global IDs is included.

There is the option to go back and cancel the whole export process by hitting the "Cancel" button. Once you have selected what is to be exported, click the "Export" button.

An iDashboards archive file is then created with the exported charts, dashboards and picklists. By default, this archive is named *idbarchive_yyyymmddhhmmssSSS.zip*. The name of the archive file may also be changed after the file is created and saved, but do not change the .zip file extension.

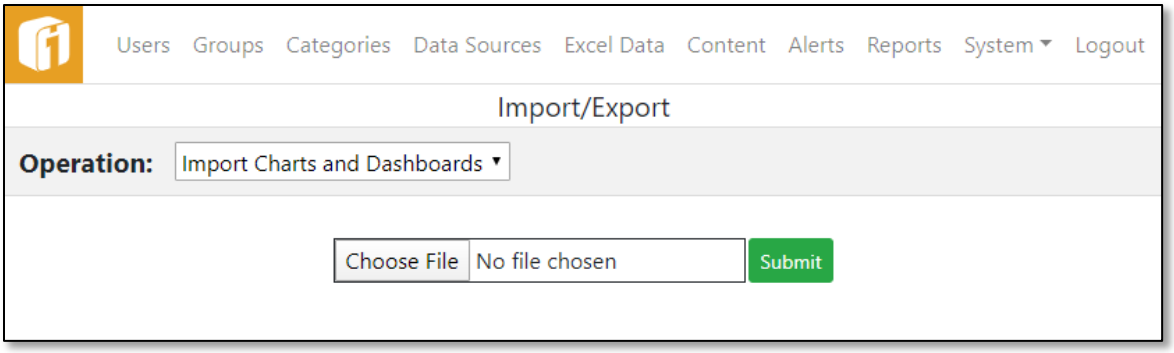
13.10.6 Importing Items

When an administrator selects "Import Items" from the "Operation" dropdown box, they will see a "Choose File" button and a "Submit" button. The "Choose File" button will assist in finding and filling out the text box which is the path of the .zip file. The administrator must do this on the iDashboards implementation that contains the target repository.

Note: If the Forms feature is enabled, within the iDashboards license, importing forms is also included.

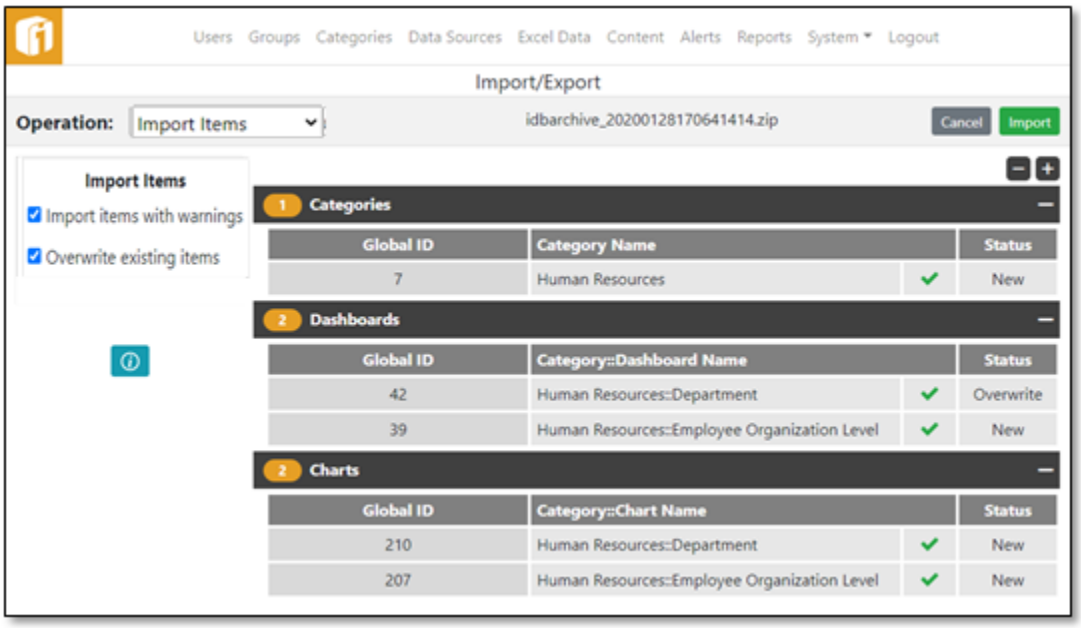
Be Aware: *When a form is first imported, its permissions are imported along with it. If a permission is "By Group", then it will have no individual group permissions. Those will have to be set manually.*

For importing a Dashboard with a "Create New" Form and a Chart for that Form's data store, the Form gets a new table on the import to data store. The Chart uses the original table. So the Chart needs to be updated to use the new table.

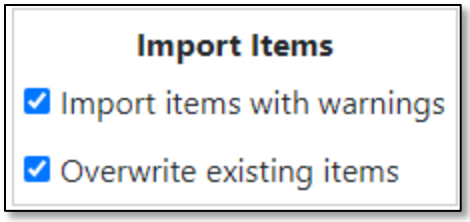


The import functionality will create all of the entities from the archive file (categories, dashboards, charts, picklists, stored procedures and forms) and assign them a GID that matches the GIDs that were assigned to the entities when they were exported from the source repository. The import function ***will not*** create any Data Sources nor assign any Data Source GIDs because any Data Source that an imported chart will use should already be created in the target repository (see Section 13.8.2, "Criteria for Import").

A list of the charts, dashboards, picklists and forms (if enabled) in the archive to be imported will be displayed. Select the ⓘ icon to view information about the import file.



Charts, dashboards, picklists and forms (if enabled) will be validated before import. The status of the validation will be displayed on the staging window. If an item is valid for import then a ✔ will show next to the status on the staging window. If there are any problems importing an item, there will be a ✖ in the status column. Select the ✖ to see a description of why it cannot be imported. To allow importing items with warnings, under "Import Options" select the box "Import items with warnings".



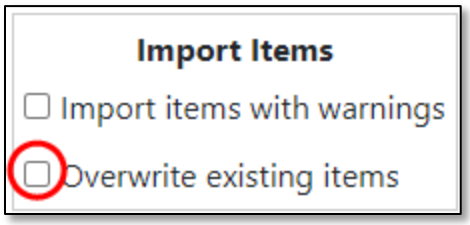
After hitting the "Import" button a confirmation message is displayed notifying you of the import into the target repository.

13.10.6.1 Importing with existing GIDs

Caution

Caution should be taken whenever importing items into a non-pristine environment. This task cannot be undone.

For example, a chart is imported with a given GID. If that GID already exists, then the import screen will display "Overwrite" in the status column. Continuing the import **will replace the existing chart with the imported chart**. If the administrator does not want to import items with matching GIDs but would like to import all other items, then, under "Import Options" unselect the box "Overwrite existing items".



Similarly, when a dashboard or picklist is imported with a given GID, if that item exists with that GID, the existing dashboard or picklist within the target repository will be replaced with the imported item. The GID of the dashboard or picklist is retained. If the item does not exist, then a new dashboard or picklist with that GID is created in the target repository.

13.10.7 Import/Export Articles

For Knowledge Base articles (if enabled), an article tree is the basic unit of import/export. When an article tree is selected for export, all of its descendant articles are included in the export, with the parent-child relationships preserved.

When an article tree is imported, any of its articles may or may not preexist in the target repository, and this determination is made by the GUID (Globally Unique ID) for each article. Non-preexisting articles are added to the target repository, while preexisting ones are overwritten. Parent-child relationships of preexisting articles may be changed or severed during the import process.

If any of the articles in an imported article tree preexist in the target repository, then the entire tree is considered an overwrite operation. If the checkbox labeled "Overwrite existing items" is not checked, then none of the articles in the tree, not even new ones, will be imported. When an imported article tree contains both new and preexisting articles, the new ones will have a strike-through "~~New~~" status rather than "New", as a hint that it will not be handled as a new item for the purposes of overwrite prevention.

When a dashboard is exported, any articles linked to it are not automatically included in the export, the way its charts are, or other dashboards linked via drilldown or dashboard launcher panels. However, if a linked article exists in the target repository, or is manually included in the export file, the link will be reestablished upon import.

In a similar fashion, dashboards linked to articles are not automatically included in the export when those articles are exported. But if they are manually included in the export, or if they preexist in the target repository (as determined by the dashboards' Global IDs) then the links will be reestablished by the import process.

13.11 Managing the License

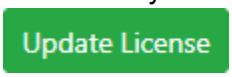
In order for the iDashboards server to function, an iDashboards license must be installed.

A new license can be installed through the Administrator application without the need to restart the server. This is done through the menu item "System > License".

The "Current License Information" page will show the status of modules as defined by the license, and the option to view and copy license details.

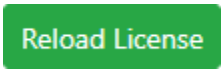
Depending on the licensing method previously used, the "Current License Information" page will have a button specific to that method.

1. License Key

   When a license key has been updated or renewed.

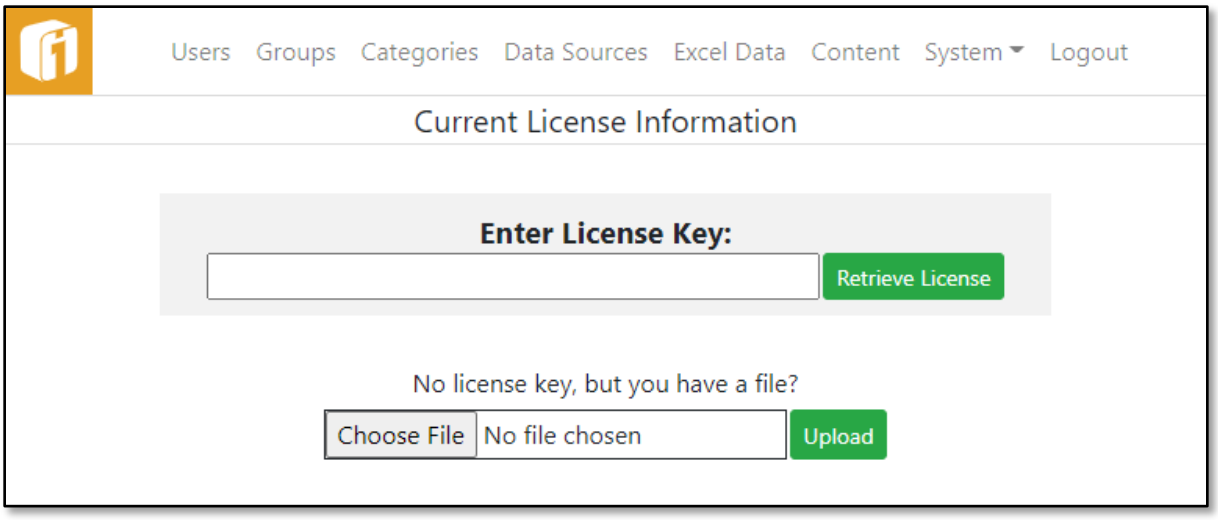
2. License File

   When a license file has been updated or renewed.

If the licensing method is being changed, select the "Change License" button 

Here the two methods to manage the license are available:



1. License Key
   a. Enter assigned License Key and select "Retrieve License" button.

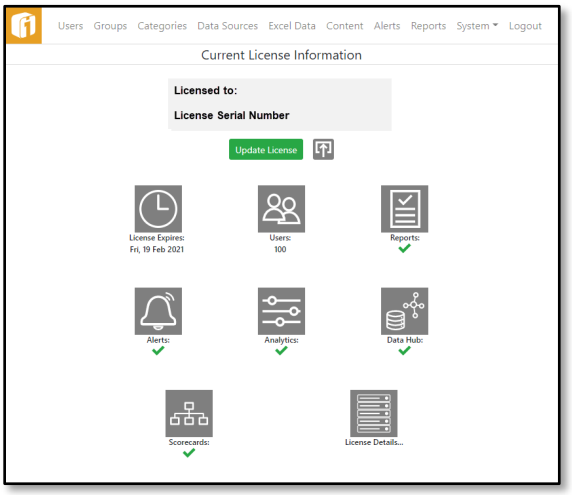
      Section 4.3 "Activating iDashboards License" covers licensing when the Administrator first logs into a new application installation.

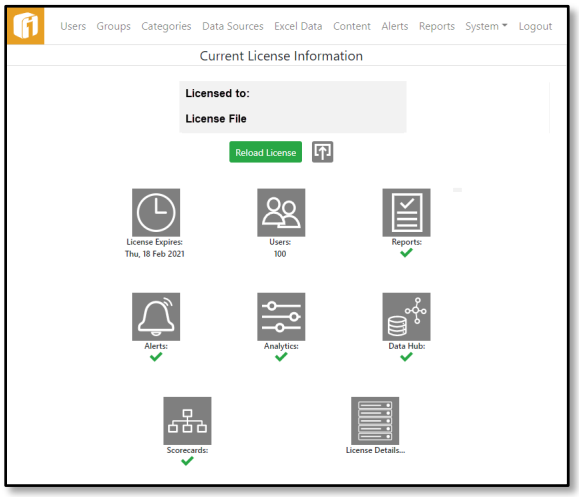
2. License File
   a. Use the "Choose File" button to locate and select the provided License File (i.e. idashboards.lic).
   b. Once selected, use the "Upload" button.

      Section 3.4.6, "Installing the License File" describes how a license file can be installed during installation of the iDashboards server.

Note: An iDashboards license file is named idashboards.lic*; however it may be named something different when initially provided. When uploaded it will be renamed to* idashboards.lic

Either way, once updated the "Current License Information" page will show the new status of modules as defined by the license, and the option to view and copy license details.

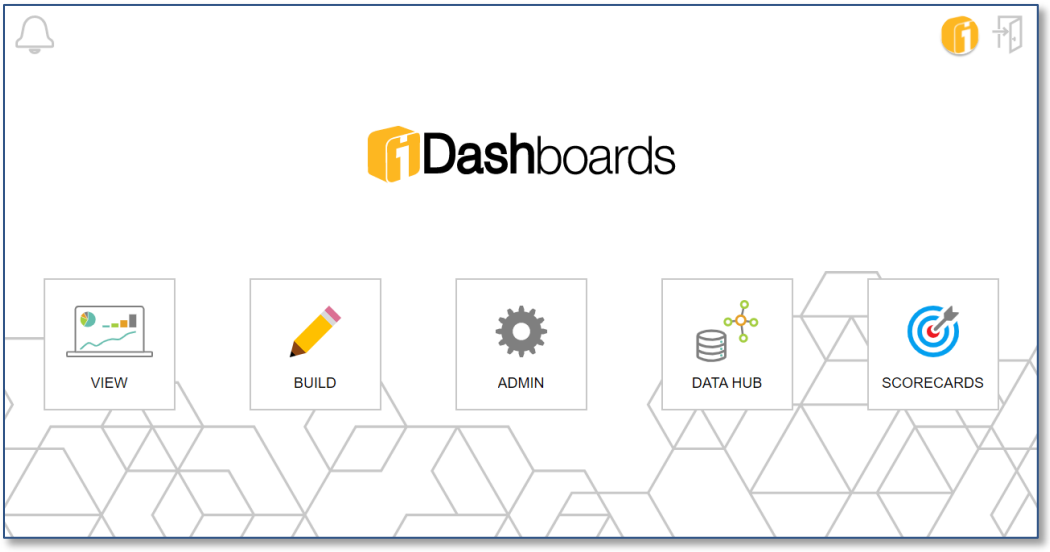




13.12 Managing the Lobby

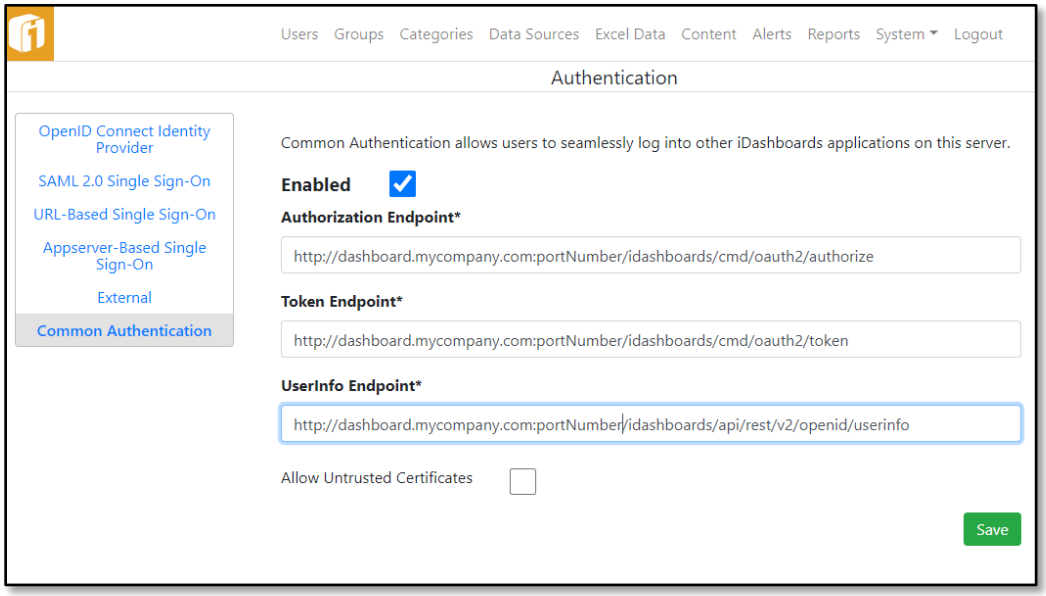
Along with the iDashboards application, Data Hub and Scorecards applications are also available. If licensed for these applications, it is possible to add them to the main iDashboards lobby. This allows a single application to login into, and provides access Data Hub and Scorecards applications without having to login again.

Note: It is necessary for Data Hub and Scorecards applications to be using the same IDB System Database as the iDashboards application.



13.12.1 Configuration

The process that makes this possible is called "Common Authentication". It is enabled through Administration using the System ▶ Authentication, and then select "Common Authentication". Use the checkbox to enable it.



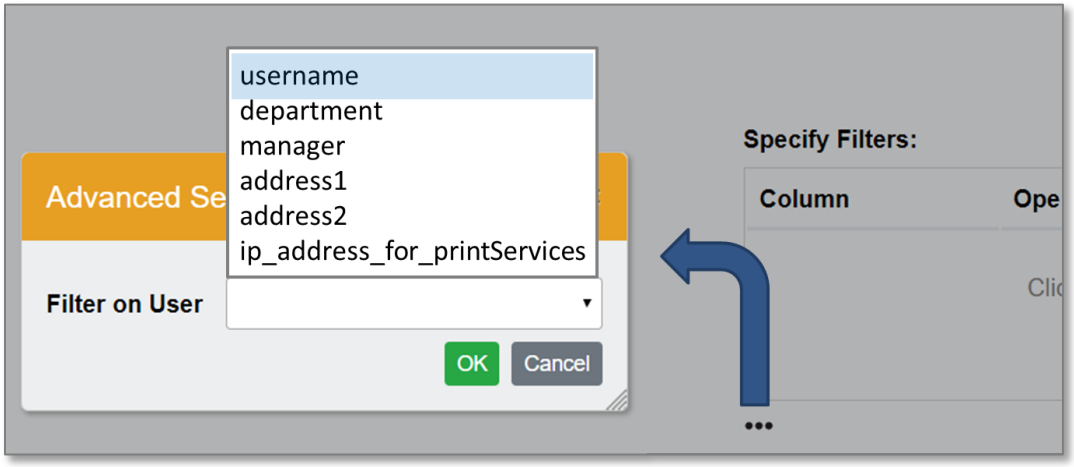
Endpoint URLs are preconfigured, but if a DNS configuration dictates it, or if there are multiple versions of iDashboards using the same repository these URLs can be modified.

If necessary, use the "Allow Untrusted Certificates" checkbox to enable that feature.

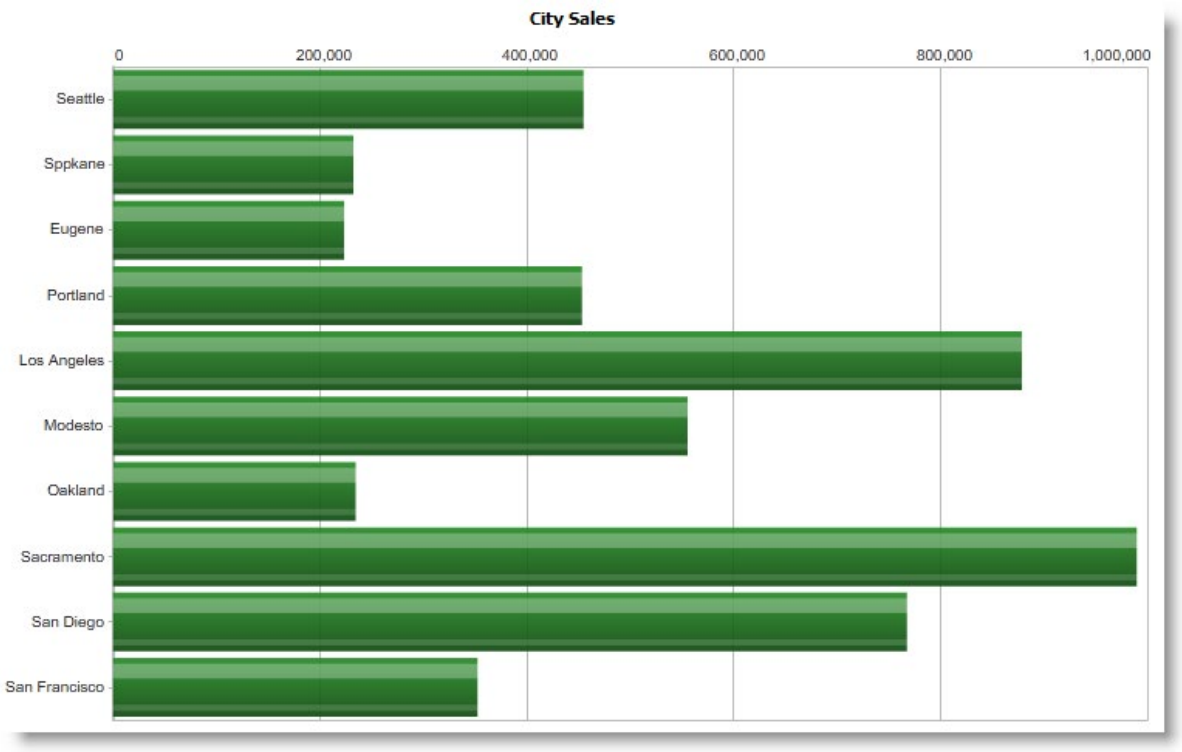
14. User Filters

User Filters are a powerful feature of iDashboards that allow the same dashboard to display different data to different users. A typical application of user filters would be in a sales organization, where sales representatives could log into iDashboards and view a dashboard of their own sales performance metrics. Another example would be a single dashboard showing regional managers charts based on data from their own regions. In each case, the user filter feature allows a single dashboard to perform the function of many, greatly reducing the time and effort required for dashboard development and maintenance.

To a Builder creating a chart through the iDashboards Build interface, user filters are an easy concept to grasp. On the screen where you might set up standard filters, the chart's data can be filtered specifically for each logged-in user by selecting a column as the "Filter on User" column. If the dropdown for "Filter-on-User" is not shown then the feature has not been configured (or was configured improperly).



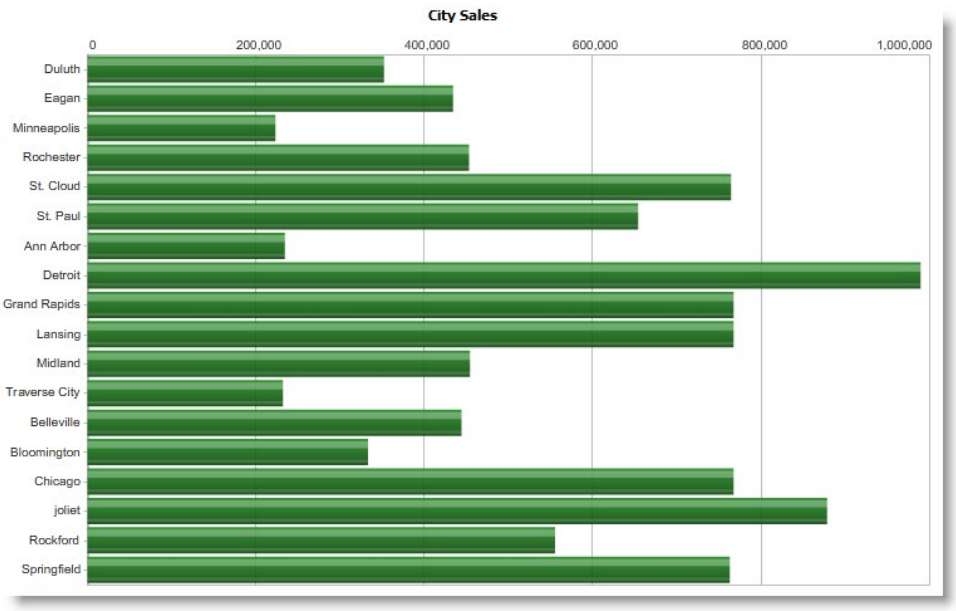
Consider this example where a chart is being constructed to display sales figures for different cities, and the STATE column is being selected as the "Filter on user" column. When a query is run to retrieve data for this chart, it will only return rows where the value of the STATE column is contained in a set of one or more STATE values configured for the logged-in user. For example, if the states configured for the username "janesmith" are "WA", "OR", and "CA", then a user logged in as "janesmith" will only see cities from Washington, Oregon and California displayed on the chart:



On the other hand, if the STATE values configured for the username "billjones" are "MN", "MI" and "IL", then a user logged in as "billjones" viewing the same chart will only see cities from Minnesota, Michigan and Illinois.

But how does iDashboards know which states are associated with "janesmith" and which are associated with "billjones"? Before the iDashboards User Filter feature can be utilized, "user filters" must be configured for applicable users in the repository database. Configuring user filters is unlike other iDashboards administrative functions in that there is no user interface provided for it in the Administrator application. (The reasons for this are discussed below.) The process used to configure and maintain user filters can be simple or complex, depending on an organization's requirements. In any case, the operations used are performed directly against the iDashboards repository database, using the administrative tools provided by the database itself, or other third-party tools. These operations should be done by, or with the assistance of, someone familiar with relational databases (preferably including the iDashboards repository database), such as a database administrator or developer.

The rest of this chapter, which discusses the configuration of user filters, assumes the reader has at least a basic understanding of relational databases and SQL (Structured Query Language).



14.1 How User Filters Work

The key component of the User Filter feature is a table in the iDashboards repository database named FV_USER_FILTER. It can have either of two structures, but in its simpler form, it will have the three columns. After creating the table, stop and start your web application service.

Column Name	Column Definition
USERNAME	VARCHAR(15) NOT NULL
COLUMN_NAME	VARCHAR(50) NOT NULL
COLUMN_VALUE	VARCHAR(255) NOT NULL

Note: fv_user_filter can be a view instead a table, but it will be referred to herein as a table.

When the FV_USER_FILTER table is populated, the USERNAME column holds the usernames (login IDs) of iDashboards users, the COLUMN_NAME column holds the names of database columns, and the COLUMN_VALUE column holds the values used to "filter" an iDashboards user's chart data.

USERNAME	COLUMN_NAME	COLUMN_VALUE
janesmith	STATE	WA
janesmith	STATE	OR
janesmith	STATE	CA
billjones	STATE	MN
billjones	STATE	MI
billjones	STATE	IL

For the examples used above to work, the FV_USER_FILTER table would have to contain the rows show above.

When performing the query for user "janesmith", the iDashboards server would do the following:

1. Check and see that the chart had the STATE column set as its "filter on user" column.
2. Read the rows from the FV_USER_FILTER table that had USERNAME set to "janesmith" and COLUMN_NAME set to "STATE". From these rows it would build a list of STATE filters for "janesmith", specifically "WA", "OR" and "CA".
3. Query the CITYSALES table (upon which the chart is based) with additional filtering criteria so that only rows with STATE columns set to "WA", "OR" or "CA" are returned.

14.2 Global vs. Table-Specific User Filters

The FV_USER_FILTER table described above tells the iDashboards server that whenever a table is queried for a chart, and the "filter on user" column is STATE and the logged-in user is "janesmith", the allowable STATE values are "WA", "OR" and "CA". This rule applies regardless of the table being queried, not just for the CITYSALES table used in the above example. User filters configured in this fashion are referred to as *global* user filters.

Having these filters apply to all tables can simplify the population and maintenance of the FV_USER_FILTER table. For example, the same FV_USER_FILTER rows that limit "janesmith"'s view of CITYSALES data to the states of Washington, Oregon and California could also be used to limit her view of WAREHOUSEINVENTORY data to those three states, provided the WAREHOUSEINVENTORY table also has a column named STATE, which contains the states where the warehouses are located.

Such simplicity comes with a cost, however, and it is not hard to imagine scenarios where global user filters would not work. Perhaps "janesmith" needs to see warehouse inventory metrics from Nevada and Arizona in addition to Washington, Oregon and California. Adding FV_USER_FILTER rows to grant her access to those states would cause them to show up in her City Sales chart as well. When such circumstances arise, it is necessary to use *table-specific* user filters rather than global user filters.

14.3 Table-Specific User Filters

Table-specific user filters work almost identically to global user filters. The key difference is that the FV_USER_FILTER contains an additional column called TABLE_NAME. The table structure would be as shown in the first table below. The TABLE_NAME column restricts each FV_USER_FILTER row to a specific database table, thus allowing separate filter sets to be defined for same-named columns on different tables.

The second table below shows how an FV_USER_FILTER table, structured for table-specific user filters, would be populated to solve the problem described in Section 14.2, "Global vs. Table-Specific User Filters". Note that for user "janesmith", whenever the

CITYSALES table is being queried, the returned data will be restricted to the states of Washington, Oregon and California, but when the WAREHOUSEINVENTORY table is being queried, the data will also include the states of Nevada and Arizona.

Column Name	Column Definition
USERNAME	VARCHAR(15) NOT NULL
TABLE_NAME	VARCHAR(15) NOT NULL
COLUMN_NAME	VARCHAR(50) NOT NULL
COLUMN_VALUE	VARCHAR(255) NOT NULL

USERNAME	TABLE_NAME	COLUMN_NAME	COLUMN_VALUE
janesmith	CITYSALES	STATE	WA
janesmith	CITYSALES	STATE	OR
janesmith	CITYSALES	STATE	CA
janesmith	WAREHOUSEINV	STATE	WA
janesmith	WAREHOUSEINV	STATE	OR
janesmith	WAREHOUSEINV	STATE	WA
janesmith	WAREHOUSEINV	STATE	AZ
janesmith	WAREHOUSEINV	STATE	NV
billjones	CITYSALES	STATE	MN
billjones	CITYSALES	STATE	MI
billjones	CITYSALES	STATE	IL
billjones	WAREHOUSEINV	STATE	MN
billjones	WAREHOUSEINV	STATE	MI
billjones	WAREHOUSEINV	STATE	IL

14.4 Strict vs. Loose User Filtering

Consider an example where a user with the username "alexp" logs into iDashboards and views the City Sales chart used in the above examples. Suppose there are no records at all on the FV_USER_FILTER table for "alexp". What cities would "alexp" see on the City Sales chart?

The default behavior would be for the City Sales chart to show *no* data to "alexp"; meaning, it would display a blank frame in the dashboard with a message saying there was no data to display. This type of behavior is called "strict" user filtering, because the appropriate FV_USER_FILTER records must exist in order for any data to be returned to the user. Strict filtering is appropriate and useful when user filtering is done for security reasons, i.e. to prevent users from viewing some data while allowing them to view other data.

In some organizations, however, user filtering may be employed as a convenience to users, rather than a security measure, to filter data out of a chart because it doesn't concern them, instead of preventing them from seeing it. Perhaps, for a given chart, there are some users who need user filters applied to a particular chart, and others for whom *no* filtering should be applied. Under a strict filtering model, this might necessitate adding numerous rows to the FV_USER_FILTER table (for example, one row for each of the 50 states for a large number of users) or creating separate but similar charts for different users. When such inconveniences arise, and the security of strict filtering is not needed, the iDashboards server can be configured to perform "loose" user filtering instead of strict user filtering. This setting, known as the "style" of user filtering, is a system-wide setting that applies to all charts employing the filter-on-user feature.

The filter-on-user style can be changed through the System Settings screen, which is described in Chapter 13, "System Configuration". Toggling the setting between strict and loose will accordingly change the filtering behavior of existing charts that have filter-on-user columns set.

The difference between strict and loose user filtering can be summarized as follows:

- **Strict Filtering + no FV_USER_FILTER records = NO data**
- **Loose Filtering + no FV_USER_FILTER records = ALL data**

14.5 Populating the FV_USER_FILTER Table

Depending on an organization's requirements, the FV_USER_FILTER table can potentially contain a large number of rows. Because of this, it is desirable to make the population and maintenance of the FV_USER_FILTER table as streamlined and automated as possible.

The information FV_USER_FILTER will contain—which departments, regions, products, etc., are associated with which users—probably already exists in a CRM (Customer Relationship Management) or ERP (Enterprise Resource Planning) system. In such cases, one possible strategy for maintaining it would be to use triggers, stored procedures or batch scripts to automatically update the FV_USER_FILTER table when information in the source system changes.

Another possible method would be to make FV_USER_FILTER a view instead of a table, based on a query of source tables visible within the repository database.

If user filtering is required for only a few charts and a few users, then the FV_USER_FILTER table can be manually maintained using tools provided by the database vendor or other third-party tools.

Given the different forms that FV_USER_FILTER can take, the large number of rows it may potentially contain, the fact that it can be a view—possibly non-updateable—rather than a table, and the fact that it may not exist at all in an iDashboards system, it should be clear why including a screen for maintaining it in the Administrator application would be impractical.

14.6 Guidelines for the FV_USER_FILTER Table or View

The following guidelines should be observed when creating and maintaining the FV_USER_FILTER table or view:

- A data connection must be made to a Table, View. Data cannot be filtered if using Custom Queries, Stored Procedures or static data.
- A FV_USER_FILTER table or view does not have to exist in the iDashboards repository database. If it does not exist, the "filter on user" dropdown will be disabled in the Chart Designer.
- All of the columns in the FV_USER_FILTER table or view should be non-null. This applies to both the global and table-specific forms.
- Although a primary key is not required in the FV_USER_FILTER table or view, it should be populated as if *all* columns form a composite primary key. In other words, no two rows on the FV_USER_FILTER table or view should be completely identical; they should all differ from the others by at least one column value.
- If the FV_USER_FILTER table or view is changed from the global structure to the table-specific structure or vice-versa, the iDashboards application server should be restarted.
- If the FV_USER_FILTER table or view is dropped or created while the iDashboards application server is running, it should be restarted.

- All of the contents of the FV_USER_FILTER table or view—usernames, column names, table names and column values—are CASE-SENSITIVE. The letter case used for column names and table names should match that shown in the Chart Data Columns screen for the corresponding objects.

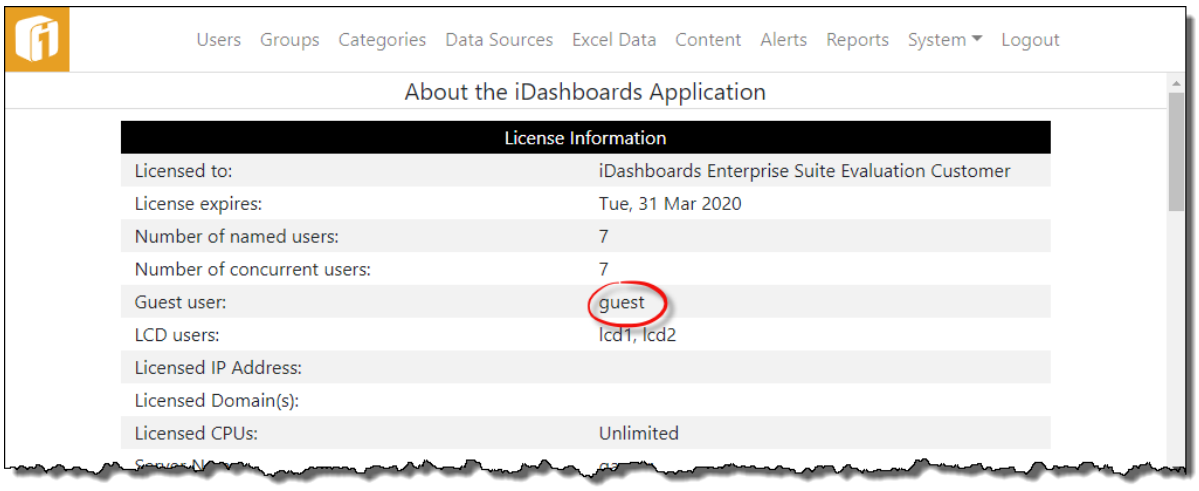
15. Guest Logins via the Public Access License

Note: The functionality described in this chapter is available only with a separately purchased Public Access License or a CPU License. An iDashboards CPU license is one for which the number of named user accounts that may be created is unlimited.

Information displayed in a dashboard is often sensitive and confidential, intended only for specific users within an organization. In some cases, however, a dashboard may be intended for viewing by everyone within a company or even by the general public over the Internet. For these occasions, the iDashboards application can run in a special mode known as "guest mode". The main differences between guest mode and normal mode are:

- A user does not need to log in with a username and password; instead a special "guest login" is performed automatically when the iDashboards application is loaded in the browser.
- Dashboards may only be viewed. They may not be modified or created.
- The Personal dashboard Category is not visible.
- User preferences may not be viewed, modified or saved.

Guest logins can only be performed for a "guest user account". A guest user account is a normal iDashboards user account that has been designated as the guest user account, and thus has the added capability of guest logins. The username of the designated guest user account is displayed on the "About iDashboards" screen of the Administrator application.



15.1 Configuring the Guest User Account

As previously mentioned, a guest user account is a normal iDashboards user account. It must be added to the iDashboards system like any other account. When a guest user account has been configured, it is identified by an asterisk (*) in the iDashboards user list.

During a guest login session, a guest user always has the privileges of the Viewer role. A guest user account can be assigned any of the four iDashboards user roles, however (Viewer, Builder or Admin) and will have all of the corresponding privileges during a normal login session. This should be taken into consideration when selecting the role for a guest user account.

A more important consideration than its user role is the user groups to which a guest user account is assigned. The categories that are visible to those groups (with the exception of the Personal Category) will determine which dashboards can be viewed during guest login sessions.

When a guest user account has been created and assigned a user role and one or more groups, the iDashboards application can be logged into normally with the guest user's login credentials, and preferences can be set and saved, such as theme color or the favorite dashboard.

15.2 Public Access License

With a Public Access License, the username of the guest user is embedded into the software license file, and cannot be changed. A user account must be created with a matching username. This license enables the "Manage Public Access" system interface for constructing a friendly guest URL.

Components of a Public Access URL –

- Name – A unique name for identify the Public Access URL.
- URL – A unique URL used for the Public Access. One can be generated using the button.
- User Name – The guest user account, used for the Public Access login.
- Mode – Embedded or Viewer. Embedded is used for embedding a chart or dashboard in another web page, such as a company intranet or portal page. Viewer uses the iDashboards' interface. The selection will control what is available to share.
- Share – Embedded mode allows sharing a dashboard or chart. Viewer mode allows a specific dashboard or access to all dashboards the guest user has permissions to view.
- Dashboard or Chart – Use 'Select…' button to choose the specific item for Public Access. Use 'Parameters…' to define those values if needed.
- Show Titlebar – For Embedded Dashboards only.
- Enable? – Unselecting the checkbox deactivates this Public Access URL.
- Expiration – A date, and time if desired, when the Public Access URL will stop working.

15.3 CPU License

With a CPU license, any user account can be designated as the guest user account through a system setting called "Guest User Account". The value of the system setting should be the username of the user account that will be used for guest logins. (See Section 13.2 "System Settings", for information on modifying a system setting.)

15.3.1 URLs for Guest Logins

A guest login is accomplished by invoking a special URL. The URL for a guest login is the same as the one used to access iDashboards, with the addition of a "guestuser" parameter. For example, if the URL normally used to access iDashboards is:

```
http://www.mycompany.com/idashboards
```

and the guest user account is "intranet", then the URL for guest logins as "intranet" would match the syntax below.

```
http://www.mycompany.com/idashboards/?guestuser=intranet
```

15.3.2 Autoloading Dashboards

If a guest user account has a favorite dashboard configured, that dashboard will be displayed automatically upon a normal guest user login. If there is no favorite dashboard configured, no dashboard will be loaded; however the user will be able to open dashboards.

A guest login URL can be modified so that any of the dashboards to which the guest user account has access will be automatically loaded upon a guest login, overriding the guest user's favorite dashboard if one has been configured. This is done by including the dashboard ID in the guest login URL, as a parameter named "dashID". For example, the following URL would automatically load the dashboard with ID 33:

```
http://www.mycompany.com/idashboards/?guestuser=intranet&dashID=33
```

The dashboard ID number is the primary key used to identify a dashboard in the iDashboards repository database. The dashboard ID for an individual dashboard is displayed on its Extended Dashboard Properties dialog in the iDashboards Build interface.

15.3.3 Embedded Viewer Mode

"Embedded viewer mode" is used for embedding a chart or dashboard in another web page, such as a company intranet or portal page. A guest user account is required for it to function. This mode is designed to enforce the sharing of a single chart or dashboard and removing the ability to freely navigate additional charts and dashboards. The only way to open additional charts or dashboards would be through the use of drill-downs or the Dashboard Launcher panel.

If the URL for accessing the iDashboards Application normally is:

```
http://www.mycompany.com/idashboards
```

Then an example URL for accessing it in embedded viewer mode, under the guest account "intranet" would be:

http://www.mycompany.com/idashboards/viewer?guestuser=intranet&dashID=33

Instead of a dashID parameter, a chartID parameter, representing the chart ID of the chart to which the guest user account has view permission, may also be used, for example:

http://www.mycompany.com/idashboards/viewer?guestuser=intranet&chartID=106

When a chartID parameter is used, the indicated chart will fill out the entire iDashboards interface.

The interface can be embedded in another HTML page through use of an IFRAME tag, for example:

<iframe style= "width:600;height:400;float:right;"
src="http://intranet/idashboards/viewer?guestuser=guest1&dashID=33"></iframe>

16. LCD Slideshow (Wall Display)

The functionality described in this chapter is available only with a separately purchased Wall Display account. Enabling this account type requires a new license from iDashboards.

Some situations may call for one, or multiple, dashboards to be displayed on an LCD monitor and have no need for a user to interact with the dashboard(s). Also, it may be necessary to cycle through multiple dashboards at a certain interval on the LCD monitor. Furthermore, it may be necessary to do this for multiple LCD monitors throughout an organization; it is called an "LCD Slideshow" and the functionality exists within iDashboards.

> *Note: LCD stands for "Liquid Crystal Display". Along with Plasma and CRT monitors, these make up the most common monitor/TV types. For the purposes of this chapter and iDashboards functionality, the term LCD is used. However, Plasma or CRT could equally be used as the "LCD Slideshow" will also work with those monitor/TV types*

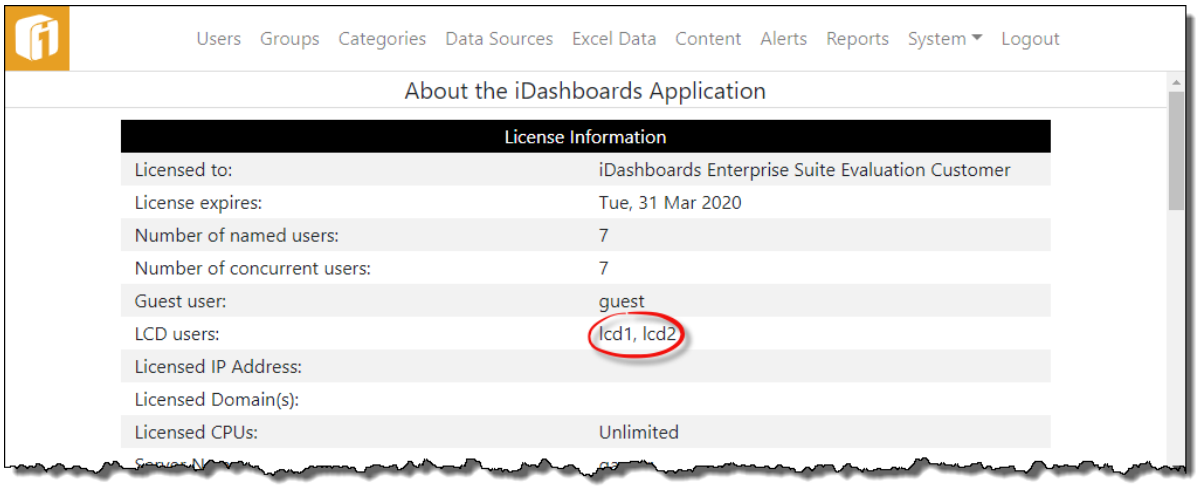
To view the LCD Slideshow, you need to log in to the specific URL using an LCD user account. When you do this, the iDashboards application will run in a special mode known as "LCD mode". The main differences between LCD mode and normal mode are:

- A user does not need to log in with a username and password; instead a special LCD login is performed automatically when the iDashboards application is loaded in the browser.
- Dashboards may only be viewed. They may not be modified or created.
- The Personal Category, and related dashboards, are not visible.
- User preferences may not be viewed, modified or saved.

> *TIP: There should be consideration on the type of dashboards that appear in the LCD cycle. Some dashboards may be designed to have very little user interaction. Other dashboards might have been designed specifically for user interaction. For example, input parameters on the chart or dashboard might require data entry before displaying. If a dashboard requires user interaction then a mouse or keyboard, or touchscreen should be provided to the user who is viewing the LCD dashboard(s) - keep this in mind when designing or selecting dashboards for an LCD Slideshow and inform the users who design the dashboards.*

16.1 LCD User Account

An LCD user account is similar to a normal iDashboards user account. It must be added to the iDashboards system like any other account. However, an LCD user account must also be specifically identified in the iDashboards license. This will be done when iDashboards is purchased with a Display License. The username(s) of the designated LCD user account(s) are displayed on menu item "System > About".



An LCD user account can be assigned any of the iDashboards user roles however (Viewer, Builder, Data Admin or Admin) and will have all of the corresponding privileges during a normal login session. However, during an LCD Slideshow, the LCD user will only have the privilege of the Viewer role. This should be taken into consideration when selecting the role for an LCD user account.

A more important consideration than its user role is the user groups to which an LCD user account is assigned. The categories that are visible to those groups (with the exception of the Personal Category) will determine which dashboards (as specified in the URL) can be viewed during the LCD mode login session. If the LCD user account does not have permissions to a certain category, the slideshow will not be able to display any of the dashboards from that category.

When an LCD user account has been created and assigned a user role to one or more groups, the iDashboards application can be logged into normally with the LCD user's login credentials, and preferences can be set and saved, such as theme color.

16.2 LCD User Account Locking

> *Note: For this section only, it should be assumed that all logins are in LCD mode versus standard login mode, meaning that the special LCD Slideshow URL was used to login and an LCD user account was used to initiate the LCD Slideshow.*

A unique functional difference between an LCD user account and a named user account is that an LCD user account gets locked to the computer when it logs into iDashboards. The LCD computer would typically be connected directly to the LCD monitor that will be used to display dashboards. Any attempts to log into iDashboards from a second computer after the LCD user account is already logged in from the first computer will result in a login error message on the second computer and a failed login attempt.

LCD user accounts get locked to a computer accessing iDashboards on a "first come, first server" basis, meaning the first computer to access iDashboards using a specific LCD user account will lock the LCD user account. Every time the iDashboards server starts up, every LCD user account is unlocked.

IMPORTANT!
To release a lock on an LCD user account, in the Slideshow Manager, select the active lock () icon next to the LCD user name. It will take 10 minutes for the lock to release.

16.3 Create and Manage LCD Slideshows

The first step is to create the LCD user accounts (See Section 16.1, "LCD User Account"). Next, go to menu item "System > Manage Slideshows". The left part of the screen will display all of the LCD users that have been created (Not necessarily all of the LCD users that can potentially be created. The potential number of LCD users, and the user names, that *can* be created is provided on the menu item "System > About"). The right part of the screen displays the slideshows. The goal is to create a slideshow and associate an LCD user (limited by the licensing) with the slideshow (no limits).

To create a Slideshow:

1. Select "New" button
2. Enter the "Slideshow Name"
3. **Show Controls**: If enabled, the slideshow control buttons will stay visible in the upper left corner of the dashboard. If disabled, the controls become visible when you move the mouse cursor over the dashboard.
4. **Show Countdown**: If enabled, the countdown timer will be displayed. If disabled, the countdown timer will not be displayed.
5. **Interval Seconds**: This is the number of seconds a dashboard in a slideshow will be displayed until it cycles to the next dashboard. The minimum is 10, and if omitted, the default is 60. Even if there is only one dashboard in the slideshow, it will still be redisplayed after the indicated number of seconds. This value will be used for all dashboards that are not individually assigned a specific "Interval Seconds" value.
6. **Refresh Minutes**: This is the number of minutes before the webpage/slideshow will restart. The purpose of using this is to ensure the browser cache is cleared occasionally. When the timeframe has been reached, the active dashboard countdown will be abandoned and the LCD cycle will begin from the first defined dashboard.
7. **Transition**: Transitions are the animations used when switching from the current dashboard to the next dashboard. The possible values are "revolve" (default), "slide", "fade" and "none". This value will be used for all dashboards in the slideshow list.

8. **Dashboards**: Navigate the list by categories. Categories are expanded with all dashboards within the category displayed in a list below. When a category is selected the dashboard list is toggled between expanded and collapsed.
9. **Slideshow**: This is the list of dashboards to be constructed. The sorted order of dashboards in the "Slideshow" list are the order in which the dashboards will be displayed when viewing the LCD Slideshow.
10. **Add, Remove and Sort Dashboards**: Highlight a dashboard, then use the Left/Right arrows to add or remove the dashboard to the Slideshow list. Within the Slideshow list, select and drag a dashboard to determine the order.
11. When finished, select "Save".

To edit a Slideshow, select its Edit icon ().

To delete a Slideshow, select its Delete icon ().

16.4 Deploying an LCD Slideshow

To deploy an LCD slideshow, you need to set the URL in a browser which has a connection to the iDashboards server hosting the dashboards. The URL is unique when compared to other methods of accessing iDashboards.

IF the URL for accessing the iDashboards application normally is:

```
http://www.mycompany.com/idashboards
```

THEN some example URL's for accessing the LCD Slideshow are:

```
http://www.mycompany.com/idashboards/lcd/

http://www.mycompany.com/idashboards/lcd/?user=lcduser1

http://www.mycompany.com/idashboards/lcd/?user=lcduser1&lrm=720
```

16.4.1 URL Parameters

This section describes how to construct, or enhance, the URL that is deployed.

user – This is the username of the account which will be used to access the server. It should be a valid LCD user account. **If the user parameter is omitted**, the server will determine which account to use by performing the following checks in the following order:

1. If a single LCD user account exists, it will be used.
2. If multiple LCD user accounts exist, then the list will be sorted in case-insensitive alphabetical order, and the first one in the list will be used.

If all of the above checks fail to produce a default username to use, or the one provided through the user parameter is not an LCD user account, an error message will be displayed. An example of user parameter assuming the LCD username is "lcduser1":

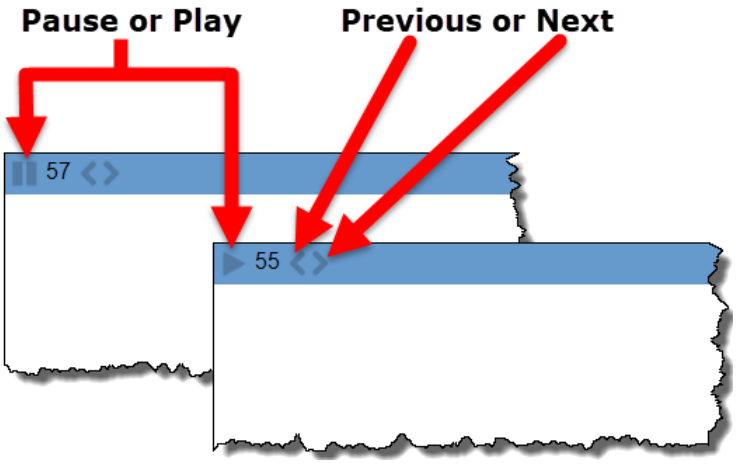
```
user=lcduser1
```

lrm – The parameter "lrm" defines the number of minutes between webpage/slideshow reloads. The purpose of using this is to ensure the browser cache is cleared occasionally. When the "lrm" is triggered, the current dashboard countdown will be abandoned and the LCD cycle will begin from the first defined dashboard. If the LCD slideshow is continuously running 24/7, it is suggested to use this parameter once a day (1440 minutes). The default is not to refresh.

```
lrm=1440
```

16.5 Interacting with an LCD Slideshow

The LCD slideshow is an automated function. While it is possible to interact with the dashboards shown during the slideshow (by using the mouse cursor), it is also possible to interact with the slideshow itself using control buttons.



16.5.1 Show Controls

If the slideshow control buttons are not visible, the buttons can still be displayed by moving the mouse cursor anywhere on the dashboard. This will present the Slideshow control buttons.

16.5.2 Slideshow Control

The LCD slideshow will initially be in "Play" mode when the URL is placed into the browser. However, if there is a need to pause (then resume) the slideshow, select the pause or play button. At any time, the mouse cursor can interact with the slideshow controls to display the next dashboard or the previous dashboard.

17. Alerts

iDashboards offers real-time alerts which can automatically trigger notifications. Alerts can be triggered based on a variety of thresholds, trend-based conditions and other criteria. When an alert is triggered, users can receive additional information at the time of the alert to assist with faster root cause analysis and problem resolution.

iDashboards Alerts is a server process which monitors chart data and awaits for the moment when a certain condition is encountered (ex. exceeding a threshold, results are blank, etc.). Once the condition is met, an alert is sent to specific iDashboards users via email or mobile SMS text message. Alerts are configured with a monitoring schedule, and will monitor the chart data even if nobody is viewing a dashboard.

The term "alert" can have different meanings in different contexts. At a general level, an alert is a mechanism that notifies iDashboards users that certain conditions exist within chart data. The term "alert" is sometimes used to refer to the notification itself, i.e. the item appearing in a user's alerts dashboard. It is also used to describe the configuration stored in the repository that defines the conditions for an alert, its name and the message that is displayed to users when the conditions are met.

Alert Terminology:

- **Alert** – Unless its context suggests otherwise, the term "alert," as used in this manual, will refer to the configuration of an alert – the conditions, name, severity level, message text, etc. – that is stored in the iDashboards repository database.
- **Check** – An alert is "checked" by the alerts server according to a predefined schedule. This means that the alerts server loads the data of the chart for which the alert was configured, and evaluates it according to the alert's rules.
- **Trigger** – If, during an alert check, the alert's rules are satisfied by the chart data, then the alert is said to be "triggered".
- **Instance** – When an alert triggers, an "instance" of the alert is created. It is this instance that appears in the alerts dashboard of the iDashboards Application.

The primary components of alerting in iDashboards involve:

- **Designing an alert** – Alerts can be configured on any chart and must be associated to a chart. You cannot have an alert without a chart. This task involves identifying the condition(s) needed to trigger an alert.
- **Scheduling an alert**– Determine the frequency the alert should check for a condition. Administrators should assist with determining an appropriate schedule since alert checking utilizes server performance.
- **Determining the audience** – Alerts can be for personal use or for groups of iDashboards users.
- **Reactivation of an alert** – Think "snooze button". Once an alert has triggered, how much time needs to pass before checking the condition again.

17.1 Alerts System

Within the regular iDashboards application, nothing much happens unless a user or administrator does something in a browser that causes a request to be sent to the server, like opening a dashboard or saving a chart. Otherwise, it sits idle, waiting for user input.

The Alerts Server is different. Even when there are no administrators logged in, the server can be busy, checking alerts, reacting to a triggered alert, sending emails, etc.

17.1.1 Alerts System Settings

Multiple settings of the Alerts Server can be controlled through the system settings screens of the iDashboards Administrator application.

See Section 13.2 "System Settings"

The four categories of system settings are:

1. 13.2.4 "SMTP Settings"
2. 13.2.7 "Alert Settings"
3. 13.2.8 "Alerts: Mobile Settings"
4. 13.2.9 "Alerts: Notification Email Settings'

17.1.2 Controlling Permissions

A user must have access to a chart before configuring an alert on a chart. Access is controlled through the permission settings for groups and categories.

17.1.3 Configure the Notification Email Settings

The iDashboards Alerts Server is capable of sending emails in response to certain events. The three types of emails sent are:

- **Alert Notifications** – An alert can be configured so that an email is sent to its recipients when the alert is triggered.
- **Server Event Notifications** – These emails are sent to a predefined list of email addresses (which presumably belong to server administrators) when certain routine (non-error) server events occur, such as the startup or shutdown of the server.
- **Server Error Notifications** – These emails are sent when certain types of errors occur on the server, such as a database error during an alert check. They are sent to the same email addresses that receive server event notifications.

17.1.4 Email Configuration Roadmap

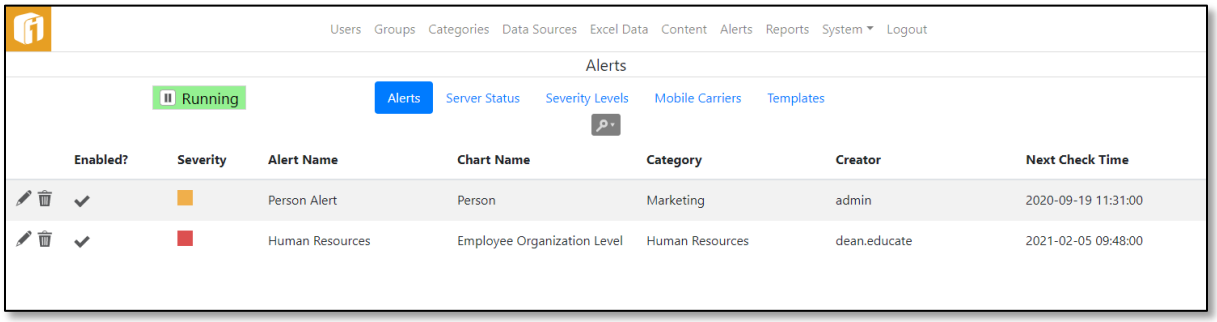
For the Alerts Server to send email notifications, it must first be properly configured. The overall steps to accomplish this are:

- **Configure the SMTP (Simple Mail Transfer Protocol) Settings** – Notification emails are sent through an external SMTP service, such as UNIX Sendmail or Microsoft Exchange Server. The Alerts Server must be configured with enough information to connect to, and if necessary, authenticate itself to the SMTP service.

- **Configure the Notification Email Settings** – These settings include information such as the name and email address used in the "from" header of outgoing emails, the list of email addresses that will receive server event notifications, and the information that is included in the subject lines of notification emails.
- **Configure the Email Templates** – This is an optional step that provides a great deal of control over the information included in the bodies of notification emails. Using email templates, notification emails can be sent in both HTML format (including images) and plain text. If this step is omitted, emails will be sent as plain text and include only a minimal amount of default information.

17.2 Alert Administration

The Alert Administration screens are accessed by clicking "ALERTS" on the Administrator application Home screen, or "Alerts" from the application menu. This will display the Alerts console along with options for other Alert Administration functions.

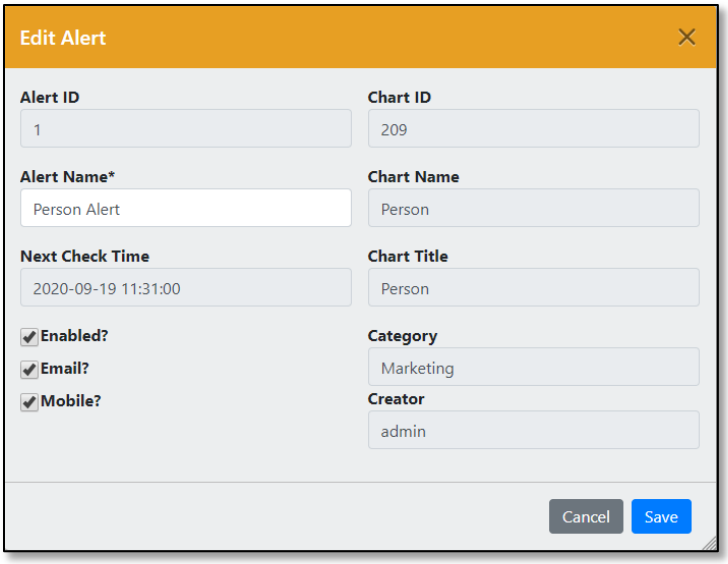


17.2.1 Alerts

Alerts are normally created and maintained through the iDashboards Application as described in the Builder Manual. Alerts under Administration, however, provides options through which limited modifications can be made to existing alerts, specifically:

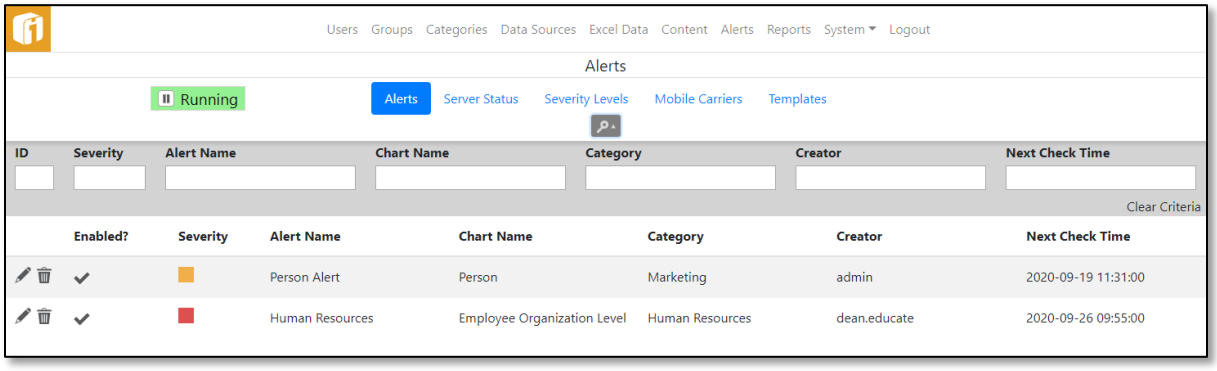
- Alerts can be enabled or disabled. When disabled, an alert is not checked by the Alerts Server.
- Email and mobile SMS text notifications can be enabled or disabled for individual alerts.
- Alerts can be deleted.

However, some details of the alert can only be edited through the User Application.



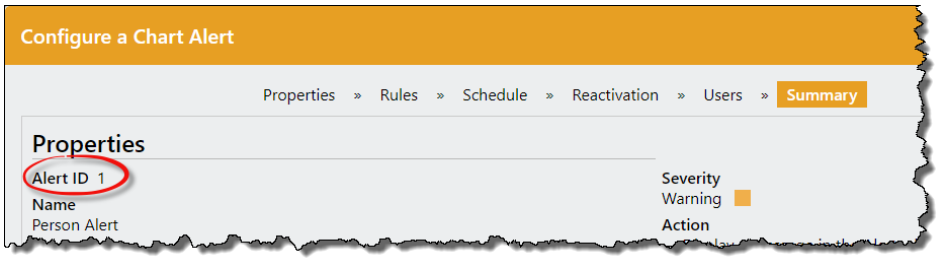
17.2.1.1 Finding an Alert

Before an alert can be modified, it must first be retrieved from the repository. Select the search icon ( ) to show the various fields in which to search against.



If the Alert ID number is known, it can be retrieved directly. The Alert ID is the number that uniquely identifies an alert in the iDashboards repository. To retrieve an alert with its ID, enter into the ID text box. If the alert with the given ID exists in the repository, it will be displayed on the screen.

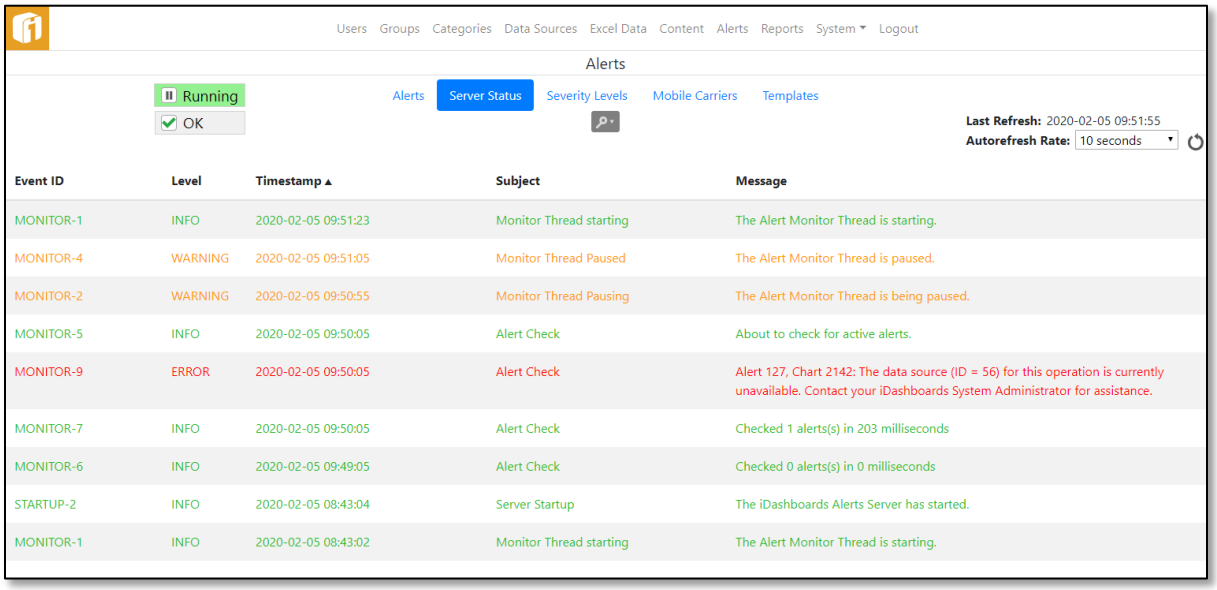
In the iDashboards Builder Application, the Alert ID is visible under Summary of the configuration dialog.



Administrative access to alerts is independent of the iDashboards security framework. An administrator can perform the above modifications on any alert in the system, regardless of the category to which the alert belongs, or whether the alert is public or private.

17.2.2 Server Status

Within the Alerts Server, an error or event can occur at any time and go unnoticed, and as a result, alerts might fail to generate when they should, or alert notifications may fail to send. The Alerts Server provides a Server Status screen through which its inner workings can be observed. To access the screen, click "Server Status" from the Alerts menu.



17.2.2.1 Pausing and Restarting the Server

At any given moment, the Alerts Server will be in one of two possible states:

- **Running** – In this state, the Alerts Server is performing all of its normal activities, such as alert checks, sending emails, etc.
- **Paused** – In this state, the Alerts Server does not perform activities such as alert checks or sending emails, however, the Alerts Server console is still fully functional.

In its default configuration, the Alerts Server enters the running state when it is started. When it is in the running state, the "State" button will show "Running". A running server can be paused by clicking the button, which will relabel it "Paused". It can be placed back into the running state by clicking the button.

Normally, the Alerts Server should be left in the running state. The paused state is generally only useful when performing troubleshooting or certain configuration changes.

17.2.2.2 Understanding Server Events

The most prominent feature of the Alerts Server Status screen is the list of server events. A server event can be any type of noteworthy occurrence, such as the server being paused, a

database error, or an alert trigger. The event list can be filtered, using the search icon (), to only display events of certain, selected levels. This is accomplished by checking or unchecking the checkboxes for the different event levels.

A server event has the following attributes:

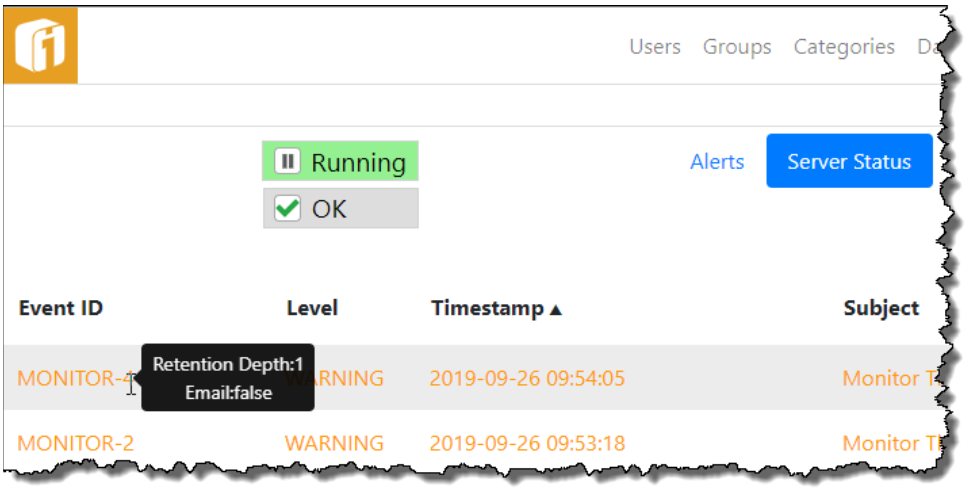
- **Event ID**
  Each server event is assigned a code referred to as the "event ID", which identifies the type of event that it is. And event ID consists of an event category, such as "MONITOR", and a number, separated by a hyphen.
  The event category is used to identify approximately where in the system the event occurred. For example, the MONITOR category is for events that occur on the monitor thread, which is the main thread that runs continually inside the server, checking alerts and performing other tasks.
  The number portion of the event ID uniquely identifies the type of event within an event category. For example, "MONITOR-7" is the event ID used to indicate that a routine alert check occurred.
- **Level**
  Each server event has one of the following three levels:
  - **INFO** – This level is used for routine events. INFO-level events are displayed in green text in the event list.
  - **WARNING** – This level is for events that occur during normal operation, but should be noted by a server administrator. WARNING-level events are displayed in the yellow text in the event list.
  - **ERROR** – This level is used for abnormal, unexpected events such as a database error that occurs during alert generation. ERROR-level events are displayed in red text in the event list.
- **Timestamp**
  The event timestamp is the date and time at which the event occurred.
- **Subject**
  The event subject is a short phrase describing the event.
- **Message**
  The event message is a short sentence that contains information about the event.

17.2.2.3 Event Retention

During normal operation, the Alerts Server is frequently recording new events in the event list. Because of this, one would expect that over time, the event list would grow extremely large, yet it does not. This is because only a certain number of events with a given event ID are retained in the event list. This number is referred to as the "retention depth" for that event ID. When the number of events with a particular event ID exceeds the retention depth for that ID, the oldest ones are removed from the list and discarded, keeping the entire event list at a manageable size.

The retention depth for an event ID is normally not of concern to the Alerts Server administrator. It can be viewed, however, by holding the mouse cursor over the event ID in

the event's list. This will produce a tool tip, similar to the one shown below, displaying the retention depth for the event ID.



17.2.2.4 Qualified Event Retention

For some error events, the retention depth is not applied to the event ID alone, but rather to the event ID combined with some hidden qualifying information. For example, if the error event is related to a particular alert, that alert's ID number might be used as the qualifying information. So, if the event ID is "MONITOR-9" and the alert ID is 123, the hidden, "qualified" event ID to which the retention depth would apply would effectively (if not actually) be "MONITOR-9-123". And if the retention depth for MONITOR-9 events is 1, that really means that one MONITOR-9 event related to alert #123 will be retained in the list, but at the same time a MONITOR-9 related to alert #905 might be retained in the list as well. This keeps important events from being pushed out of the event list before they can be viewed by an administrator.

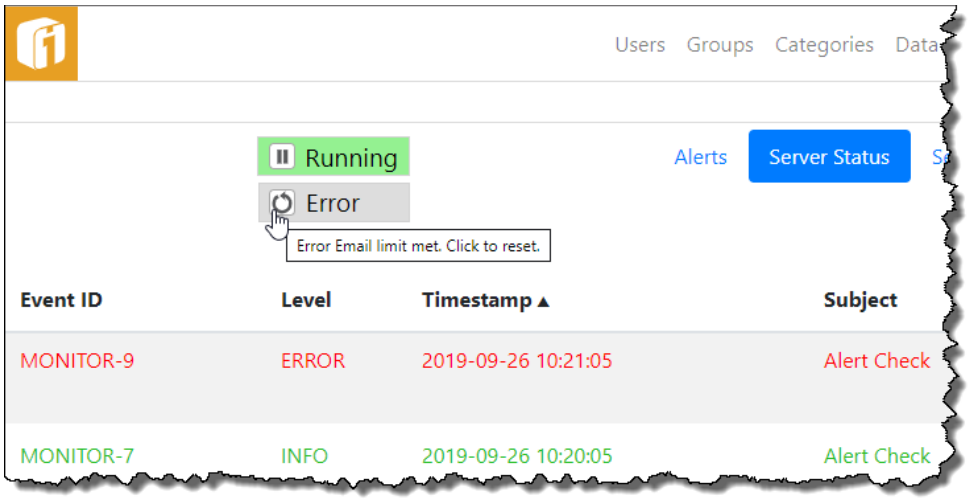
17.2.2.5 Email Events

Certain event types are designated as "email events". When an email event occurs, a notification email will be sent to the designated Alerts Server administrators, provided that:

- The Alerts Server is properly configured to send event notification emails.
- The level of the event (INFO, WARNING, or ERROR) is at or above the configured threshold at which the event notification emails are sent.

To determine whether or not an event in the list is an email event, hold the mouse cursor over its event ID until the tool tip appears. It will include the line "Email: true" for email events, and "Email: false" for non-email events.

When an ERROR qualified event occurs it will send the email and set the Email Limit button to "Error", with an "Error Email limit met. Click to reset." tooltip.



This will prevent the Alerts Server administrators from continually receiving notifications for the same issue. To begin receiving ERROR emails, select the "Error" button to reset the limit counter. This will update the button to say "OK".

17.2.3 Severity Levels

Every alert has a severity level associated with it, which indicates whether the news brought by the alert is good or bad, and to what degree it is good or bad. A severity level is represented by a name and color. The meaning associated with a severity level is determined by the administrator who configures it.

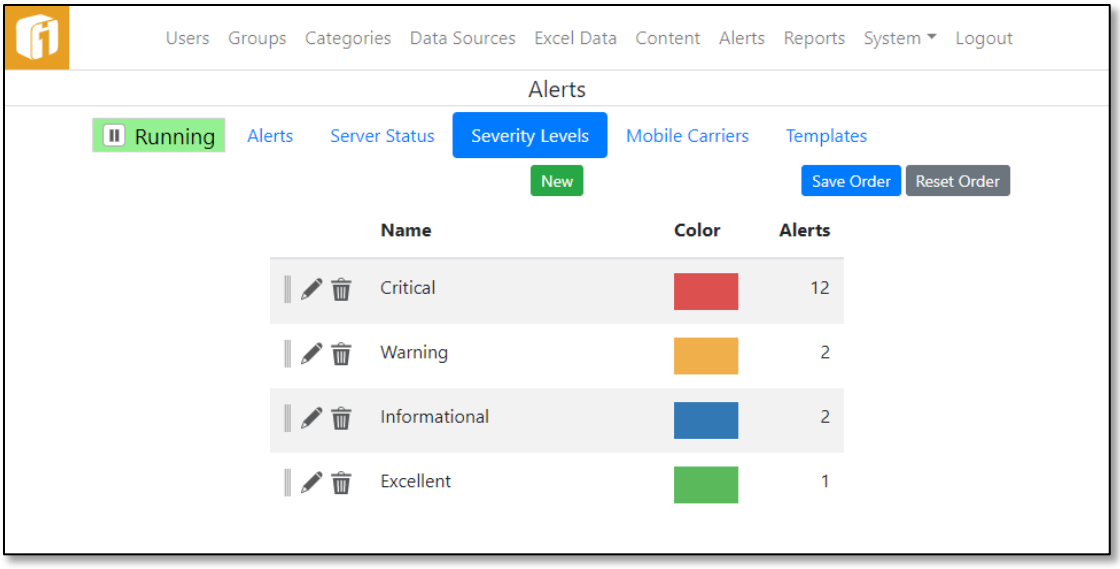
- **The Severity Name** – a short name, for example "Critical" or "Monthly Sales Goals Reached".
- **The Severity Color** – a color that is displayed on alert notifications.

In its default configuration, the Alerts Server provides four built-in severity levels:

Name	Color
Critical	**Red**
Warning	**Yellow**
Informational	**Blue**
Excellent	**Green**

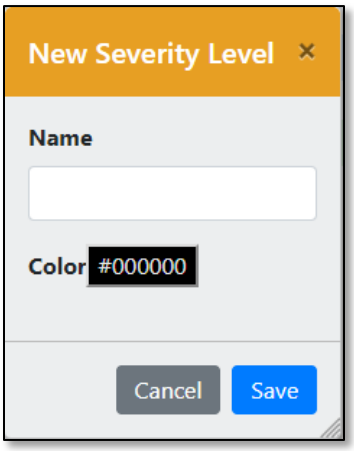
In addition to the built-in severity levels, an administrator can add new ones and delete existing ones.

Severity levels are managed through the Severity Levels screen. To access the Severity Levels screen, select Severity Levels from Alerts Administration.



17.2.3.1 Adding a Severity Level

To add a severity level click the "New" button on the Severity Levels screen. This will open the Severity Level edit screen. Enter a name consisting of from one to 50 characters. The severity color is defined by selecting Hex Code Color button, and using the Color Picker. After the New Severity Level screen has been completed, click the Save button to save the new severity level, or the Cancel button to dismiss the screen without saving it.



17.2.3.2 Modifying a Severity Level

To modify a severity level, click its Edit icon ( ) on the Severity Levels screen. This will open the Edit Severity Level screen, through which the severity level's name and color can be modified.

To save the changes, click the Save button, or click the Cancel button to discard the changes and dismiss the screen.

> *Note: Any changes made to a severity level will be visible on any alerts that have a severity level, and all instances of those alerts.*

17.2.3.3 Deleting a Severity Level

Severity levels can be deleted, provided that no existing alerts are using them as their severity level. The numbers of alerts that are using each severity level are shown in the Alerts column on the Severity Levels screen.

To delete a severity level, click its Delete icon () on the Severity Levels screen. If there are no alerts using it, it will be deleted after confirmation.

17.2.3.4 Ordering Severity Levels

Severity Levels can be listed in any order, to control how they are displayed when later used. There is the ability to alphabetically sort all Severity Levels by selecting the column title and toggling ▾ ascending or ▴ descending. To individually sort Severity Levels, Select-n-Drag it using the reorder icon () at the beginning of each row. To keeps this sort order select the "Save Order" button. To go back to the original sort order select the "Reset Order" button.

17.2.4 Mobile Carriers

Warning!
ASSOCIATED COSTS OF MOBILE TEXT MESSAGES

SMS is essentially the text messaging service offered by all major mobile carriers. The iDashboards Alerts feature has an option to send an SMS text message to users when an alert is triggered. SMS text messages are not enabled by default. The iDashboards administrator and each user partake in setting up each user enrollment in receiving SMS alerts.

Not all users subscribe to SMS or may incur a cost for all SMS text messages received. Before using Alerts with mobile text messages, review the SMS capabilities and costs of any iDashboards user configured in the system.

When each user enables Mobile notifications, they will see the following warning:

> *WARNING: By choosing to receive text messages, you may incur additional charges from your mobile phone provider.*

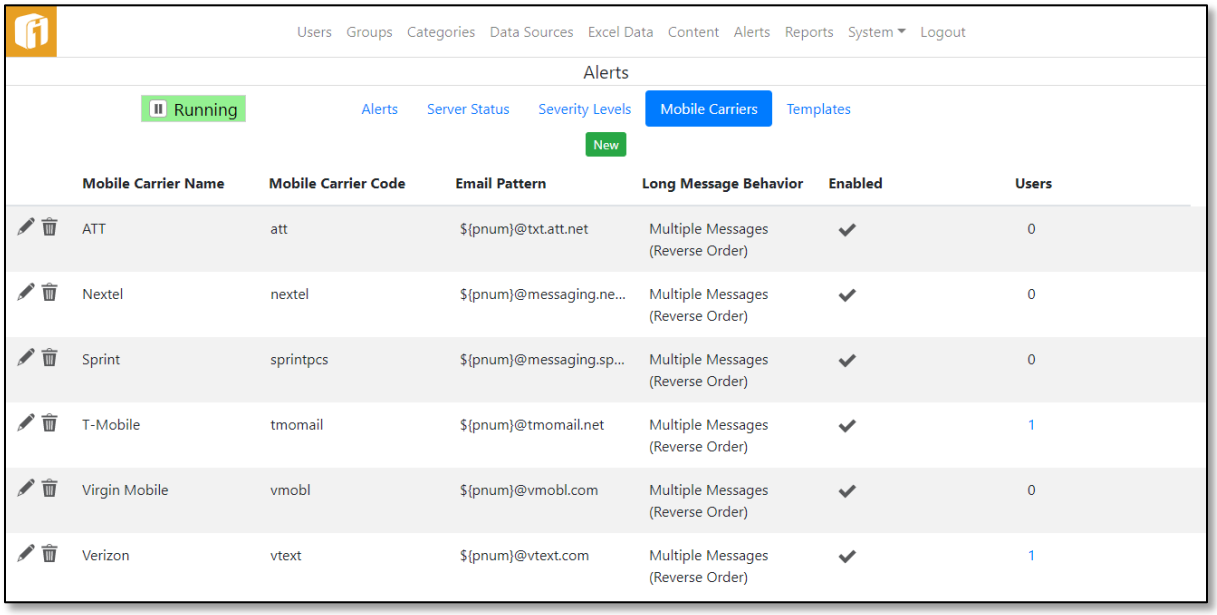
iDashboards assumes no responsibility for fees incurred by utilizing the SMS text messaging feature. Any mobile SMS text messaging fees that are incurred will be billed on 'your' individual mobile provider bill.

When each user enables receiving alert notifications by Text message, through their User Settings, they will see the following warning:

> WARNING: By choosing to receive text messages, you may incur additional charges from your mobile phone provider.

Within iDashboards, all major mobile phone provider can be configured with a unique Short Message Service (SMS) set of technical properties. Participating mobile SMS carriers include (but are not limited to): AT&T, Nextel, Sprint, T-Mobile, Virgin Mobile, and Verizon. Upon installation of Alerts, a variety of popular carriers are available. Existing carriers can be deleted or edited and additional carriers can be added. A carrier already appearing in the list may need to be added more than once to compensate for legacy SMS syntax differences.

Mobile SMS carriers are managed through the Mobile Carriers screen. To access the Mobile Carriers screen, select Mobile Carriers from Alerts Administration.



17.2.4.1 Edit a Carrier

Any carrier appearing in the list can be edited. To modify a carrier, click its Edit icon ( ). The Mobile Carrier code cannot be changed, but all other fields can be updated.

17.2.4.2 Deleting a Carrier

To delete a carrier, click its Delete icon ( ). Carriers can be deleted if the number of associated users is zero, as seen in the right hand column. If a carrier has associated users,

then the user much dissociate their User Profile with the Carrier. Users can associate or dissociate their account with a single carrier through their User Settings.

17.2.4.3 Add a Carrier

Select the "New" to create a new carrier. All fields are required to add a carrier. Each carrier created will help convert email messages into text messages using a syntax provided by your carrier.

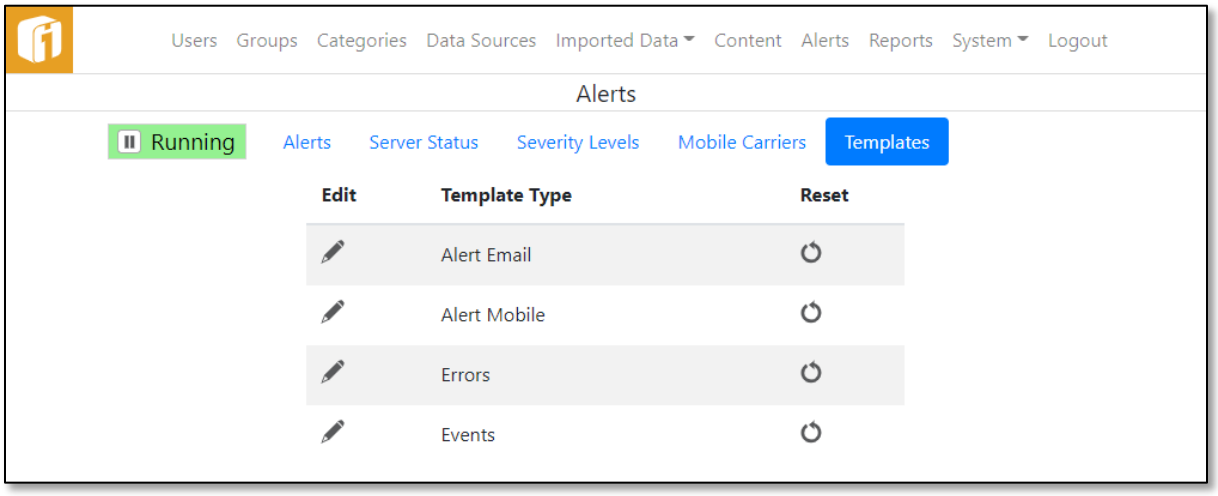
- **Mobile Carrier Name** – This field is used to display the mobile SMS carrier name in a friendly format. If a carrier has a legacy carrier code syntax you will need to create multiple entries for the carrier; however, the carrier name cannot be duplicated.
    - **Tip**: If a carrier needs to be entered more than once, try to simply add a numeric suffix (ex. AT&T_2)
- **Mobile Carrier Code** – Each mobile carrier has a unique code that is often derived from the Email Pattern below.
- **Email Pattern** – Each Mobile carrier has a unique Email-to-Text string that is capable of converting an email message into a text message.
    - **Prefix** - ${pnum}
        - The "pnum" macro stands for phone number. This macro will be programmatically replaced with a users' 10-digit phone number (ex. 2485551212@txt.att.net)
    - **Suffix -** <email pattern>
        - The first character will be the "@" symbol, followed by the email-to-text string provided by your carrier.
- **Long Message Behavior –** Each carrier has a configuration for handling long text messages. Keep in mind carriers usually have a text message limit of 160 characters.
    - **Send Entire Message –** This option will attempt to send a text message of any character length. If the character length exceeds 160, then the carrier provider will handle message segmentation.
    - **Truncate Message –** This option will ensure the iDashboards text message will be reduced at-or-below 160 characters before sending the message.
    - **Multiple Messages –** This option will have iDashboards segment any messages greater than 160 characters, sending multiple messages if necessary.
    - **Multiple Messages (Reverse Order) <default> –** Same as "Multiple Messages", but the sending order will be in reverse order (sending the last message segment first, and continuing until the first segment is sent last)
- **Enabled –** This setting will allow for globally enabling or disabling of a carrier in the list.

Note: If an entire carrier needs to be enabled or disabled, the administrator may have to enable or disable multiple entries to accommodate for multiple entries associated to legacy syntax.

17.2.5 Templates

This is an optional step that provides a great deal of control over the information included in the bodies of notification emails and mobile SMS text messages. Using templates, notification emails can be sent in both HTML format (including images) and plain text. If left untouched, notifications will be sent as plain text and include only a minimal amount of default information.

Templates are managed through the Templates screen. To access the Templates screen, select Templates from Alerts Administration.



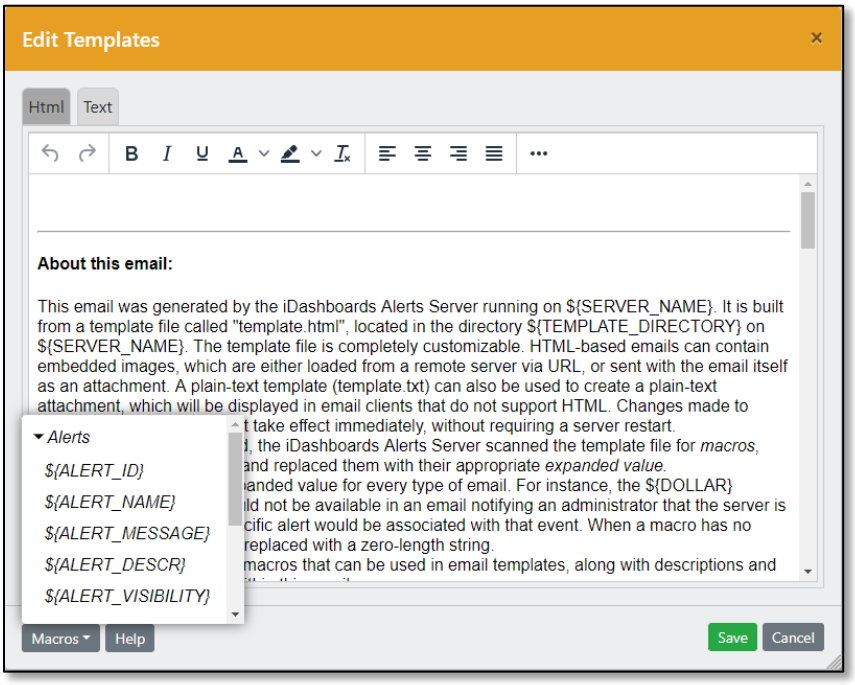
There are 4 Alerts Template Types:

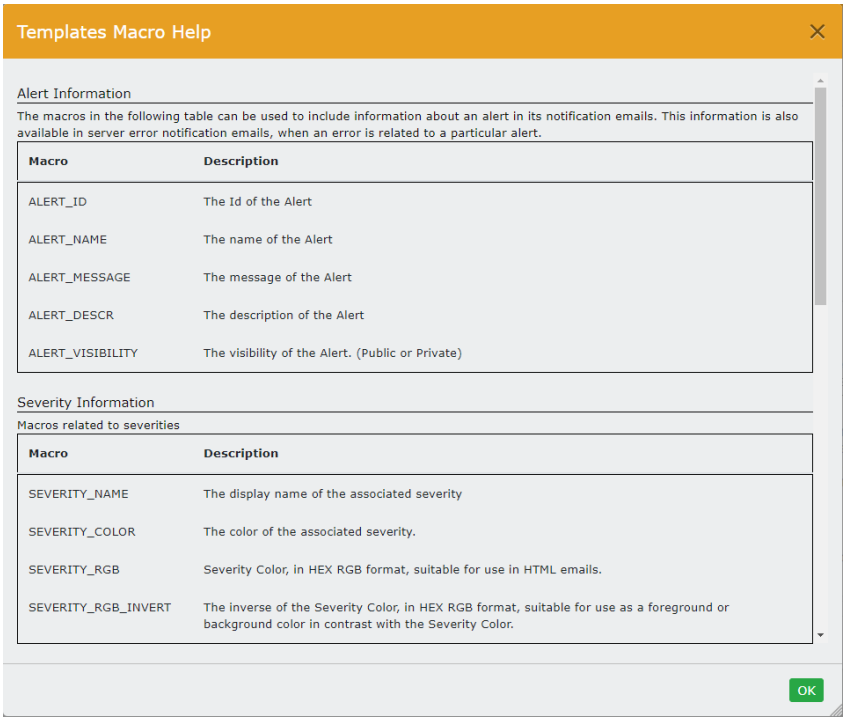
Type	Format	Usage
Alert Email	HTML/Text	Alert Notifications by Email
Alert Mobile	Text	Alert Notifications by Mobile SMS Text Message
Errors	HTML/Text	Alert Server Error Notifications by Email.
Events	HTML/Text	Alert Server Event Notifications by Email.

To modify a Template, click its Edit icon ( ). The HTML panel provides a rich text editor for formatting and laying out an HTML emails, and the Text panel has a simple plain text box for plain text messages. The first time a Template is edited the system creates sample HTML and Text notifications, with examples of using the available alert macros. Macros can found and selected by using the Macros button. The Help button will also provide details about the template macros.

After the first save of a template, the template files are created in an iDashboards' application directory, identified by the ${TEMPLATE_DIRECTORY} macro.

Using a Template's Reset icon ( ) will regenerate the system created sample HTML and Text notifications for it, and also create the template files in ${TEMPLATE_DIRECTORY}.





18. Reports

iDashboards Reports is functionality that generates data reports based upon data used to display a live chart or dashboard. The reports generated represent values from a snap-shot taken at a particular time. All of the data values from each chart are displayed all at once within a report, using a tabular, non-graphical format.

iDashboards Reports provides the following capabilities:

- On-demand report generation from the client interface.
- Automated reporting using a customized schedule.
- Security to determine who is able to generate reports.
- Report generation on a single chart or an entire dashboard.
- Option to include a customizable report cover page, with optional dashboard image.
- Customizable report page layout.

The primary components of reporting in iDashboards involve:

- **Designing a report** – Reports are automatically available on every chart and dashboard. However, there are options to specify report properties that control what information the report should display and how it should look.
- **Viewing a report** – Running a report will use default settings, or specific settings that were defined during the optional design stage.
- **Scheduling a report** – Determine if a chart or dashboard report should remain an on-demand report or an automatically scheduled report.
- **Determining the audience** – Scheduled Reports can be for personal use or for groups of iDashboards users.

18.1 Reports System

iDashboard's reports are created as PDF files, for viewing, saving and sharing.

Three roles (Admin, Data Admin, and Builder) will instantly be able to run reports on any dashboard they have access to. Viewers and Guests will have the ability to run reports if granted. See Section 13.2.5, "Report Settings" for these settings. By default, a Viewer can run a report and a Guest cannot run a report.

Users with the role of Admin, Data Admin or Builder will have full ability to create custom reporting options within their permitted categories.

18.1.1 Reports System Settings

Multiple settings of the Reports Server can be controlled through the system settings screens of the iDashboards Administrator application.

See Section 13.2 "System Settings"

The three categories of system settings are:

1. 13.2.4 "SMTP Settings"
2. 13.2.5 "Report Settings'
3. 13.2.6 "Reports: Notification Email SettingsSMTP Settings"

18.1.2 Controlling Permissions

A user must have access to a dashboard and chart before configuring a report on them. Access is controlled through the permission settings for groups and categories.

18.1.3 Configure the Notification Email Settings

The iDashboards Reports Server is capable of sending emails in response to certain events. The three types of emails sent are:

- **Report Notifications** – A report can be configured so that an email is sent to its recipients when the report's schedule is triggered.
- **Server Event Notifications** – These emails are sent to a predefined list of email addresses (which presumably belong to server administrators) when certain routine (non-error) server events occur, such as the startup or shutdown of the server.
- **Server Error Notifications** – These emails are sent when certain types of errors occur on the server, such as a database error during a reports schedule check. They are sent to the same email addresses that receive server event notifications.

18.1.4 Email Configuration Roadmap

For the Reports Server to send email notifications, it must first be properly configured. The overall steps to accomplish this are:

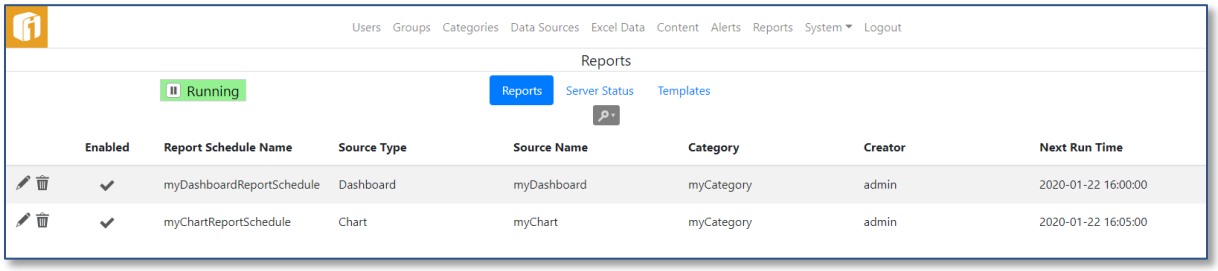
- **Configure the SMTP (Simple Mail Transfer Protocol) Settings** – Notification emails are sent through an external SMTP service, such as UNIX Sendmail or Microsoft Exchange Server. The Reports Server must be configured with enough

information to connect to, and if necessary, authenticate itself to the SMTP service (see Section 13.2.4, "SMTP Settings").

- **Configure the Notification Email Settings** – These settings include information such as the name and email address used in the "from" header of outgoing emails, the list of email addresses that will receive server event notifications, and the information that is included in the subject lines of notification emails.
- **Configure the Email Templates** – This is an optional step that provides a great deal of control over the information included in the bodies of notification emails. Using email templates, notification emails can be sent in both HTML format (including images) and plain text. If this step is omitted, emails will be sent as plain text and include only a minimal amount of default information.

18.2 Reports Administration

The Reports Administration screens are accessed by clicking "REPORTS" on the Administrator application Home screen, or "Reports" from the application menu. This will display the Reports console along with options for other Reports Administration functions.



18.2.1 Reports

Reports are normally created and maintained through the iDashboards Application as described in the Builder Manual. Reports under Administration, however, provides options through which limited modifications can be made to existing report schedules, specifically:

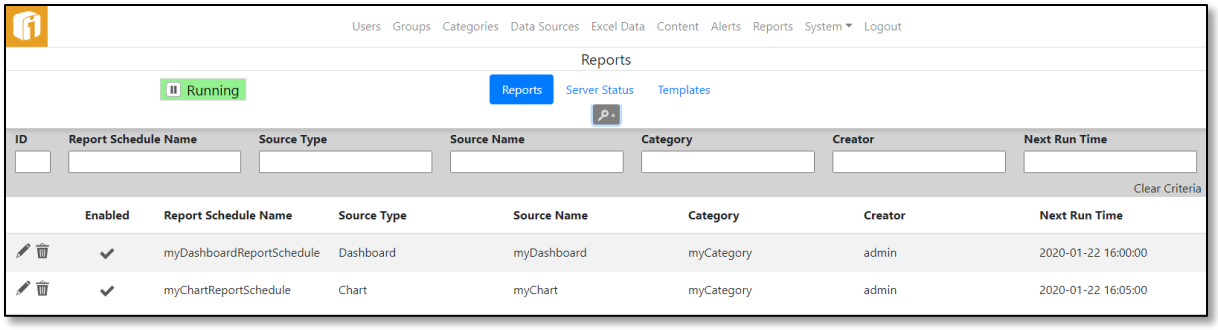
- Report Schedules can be enabled or disabled. When disabled, the schedule is not checked by the Reports Server.
- The Report Schedule name can be changed.
- Reports can be deleted.

However, some details of the report can only be edited through the User Application

18.2.1.1 Finding an Report Schedule

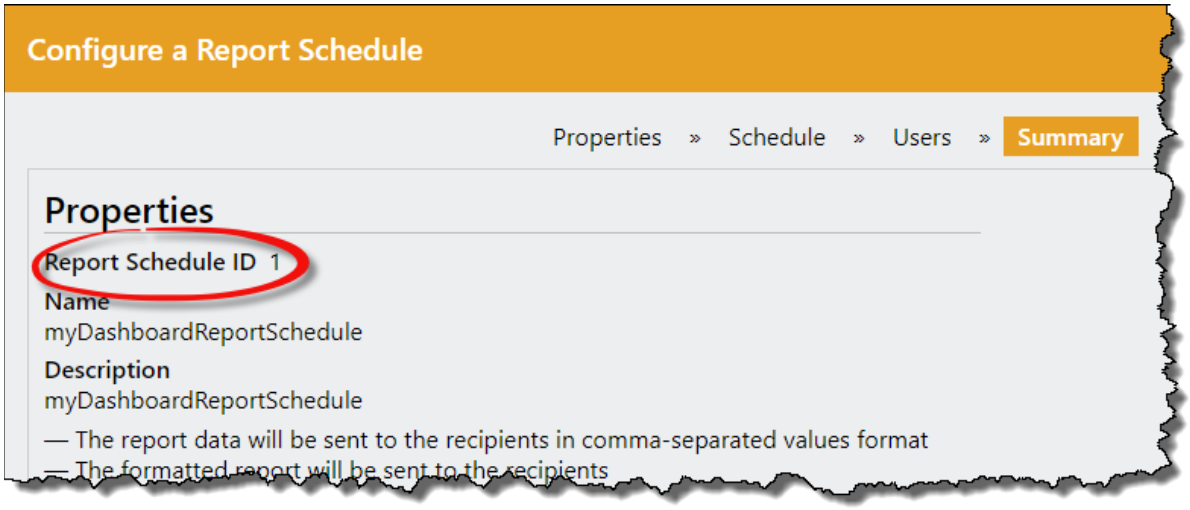
Before a report schedule can be modified, it must first be retrieved from the repository.

Select the search icon ( ) to show the various fields in which to search against.



If the Report Schedule ID number is known, it can be retrieved directly. The Report Schedule ID is the number that uniquely identifies an alert in the iDashboards repository. To retrieve an alert with its ID, enter into the ID text box. If the alert with the given ID exists in the repository, it will be displayed on the screen.

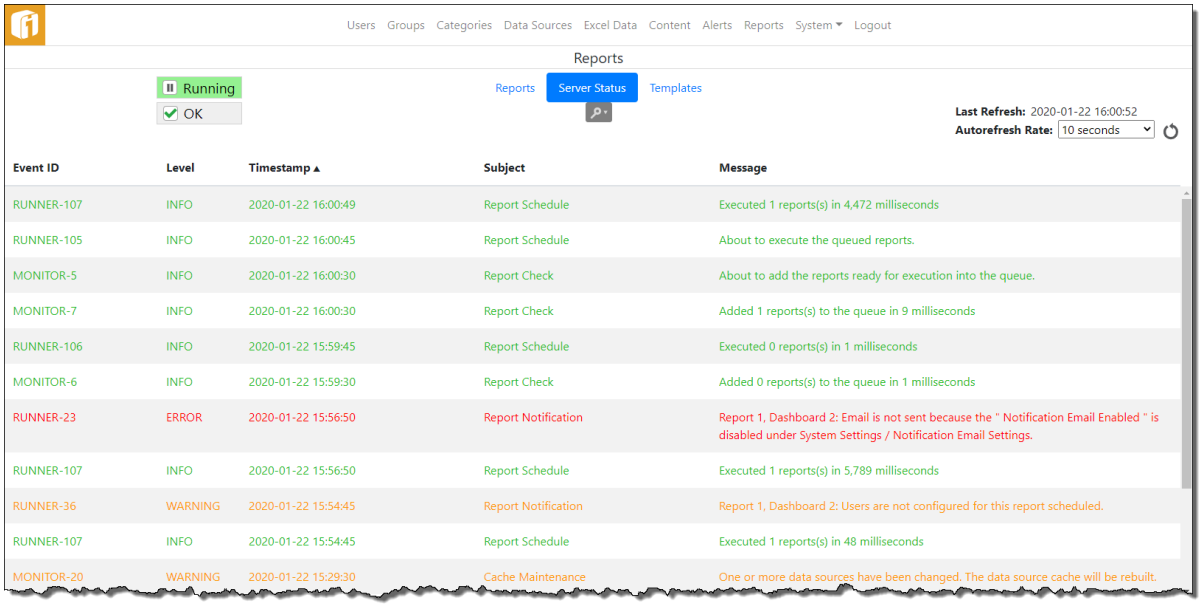
In the iDashboards Builder Application, the Report ID is visible under Summary of the configuration dialog.



Administrative access to reports is independent of the iDashboards security framework. An administrator can perform the above modifications on any report in the system, regardless of the category to which the report belongs, or whether the report is public or private.

18.2.2 Server Status

Within the Reports Server, an error or event can occur at any time and go unnoticed, and as a result, report schedules might fail to generate when they should, or report notifications may fail to send. The Reports Server provides a Server Status screen through which its inner workings can be observed. To access the screen, click "Server Status" from the Reports menu.



18.2.2.1 Pausing and Restarting the Server

At any given moment, the Reports Server will be in one of two possible states:

- **Running** – In this state, the Reports Server is performing all of its normal activities, such as schedule checks, sending emails, etc.
- **Paused** – In this state, the Reports Server does not perform activities such as schedule checks or sending emails, however, the Reports Server console is still fully functional.

In its default configuration, the Reports Server enters the running state when it is started. When it is in the running state, the "State" button will show "Running". A running server can be paused by clicking the button, which will relabel it "Paused". It can be placed back into the running state by clicking the button.

Normally, the Reports Server should be left in the running state. The paused state is generally only useful when performing troubleshooting or certain configuration changes.

18.2.2.2 Understanding Server Events

The most prominent feature of the Reports Server Status screen is the list of server events. A server event can be any type of noteworthy occurrence, such as the server being paused, a database error, or an alert trigger. The event list can be filtered, using the search icon ( ), to only display events of certain, selected levels. This is accomplished by checking or unchecking the checkboxes for the different event levels.

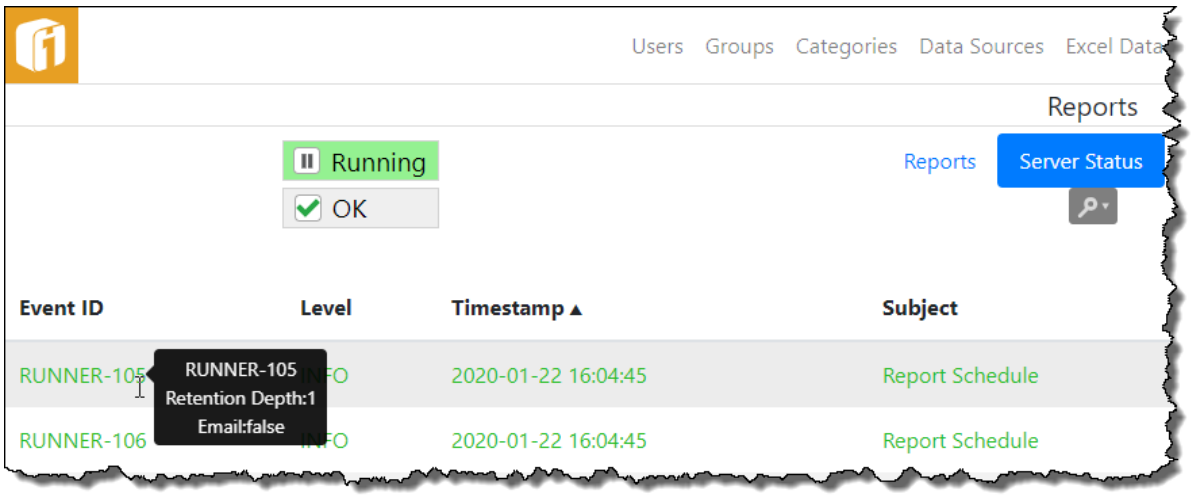
A server event has the following attributes:

- **Event ID**
  Each server event is assigned a code referred to as the "event ID", which identifies the type of event that it is. And event ID consists of an event category, such as "MONITOR", and a number, separated by a hyphen.
  The event category is used to identify approximately where in the system the event occurred. For example, the MONITOR category is for events that occur on the monitor thread, which is the main thread that runs continually inside the server, checking schedules and performing other tasks.
  The number portion of the event ID uniquely identifies the type of event within an event category. For example, "MONITOR-7" is the event ID used to indicate that a routine schedule check occurred.
- **Level**
  Each server event has one of the following three levels:
  - **INFO** – This level is used for routine events. INFO-level events are displayed in green text in the event list.
  - **WARNING** – This level is for events that occur during normal operation, but should be noted by a server administrator. WARNING-level events are displayed in the yellow text in the event list.
  - **ERROR** – This level is used for abnormal, unexpected events such as a database error that occurs during alert generation. ERROR-level events are displayed in red text in the event list.
- **Timestamp**
  The event timestamp is the date and time at which the event occurred.
- **Subject**
  The event subject is a short phrase describing the event.
- **Message**
  The event message is a short sentence that contains information about the event.

18.2.2.3 Event Retention

During normal operation, the Reports Server is frequently recording new events in the event list. Because of this, one would expect that over time, the event list would grow extremely large, yet it does not. This is because only a certain number of events with a given event ID are retained in the event list. This number is referred to as the "retention depth" for that event ID. When the number of events with a particular event ID exceeds the retention depth for that ID, the oldest ones are removed from the list and discarded, keeping the entire event list at a manageable size.

The retention depth for an event ID is normally not of concern to the Reports Server administrator. It can be viewed, however, by holding the mouse cursor over the event ID in the event's list. This will produce a tool tip displaying the retention depth for the event ID.



18.2.2.4 Qualified Event Retention

For some error events, the retention depth is not applied to the event ID alone, but rather to the event ID combined with some hidden qualifying information. For example, if the error event is related to a particular alert that report schedule's ID number might be used as the qualifying information. So, if the event ID is "MONITOR-9" and the report schedule ID is 123, the hidden, "qualified" event ID to which the retention depth would apply would effectively (if not actually) be "MONITOR-9-123". And if the retention depth for MONITOR-9 events is 1, that really means that one MONITOR-9 event related to report schedule #123 will be retained in the list, but at the same time a MONITOR-9 related to report schedule #905 might be retained in the list as well. This keeps important events from being pushed out of the event list before they can be viewed by an administrator.

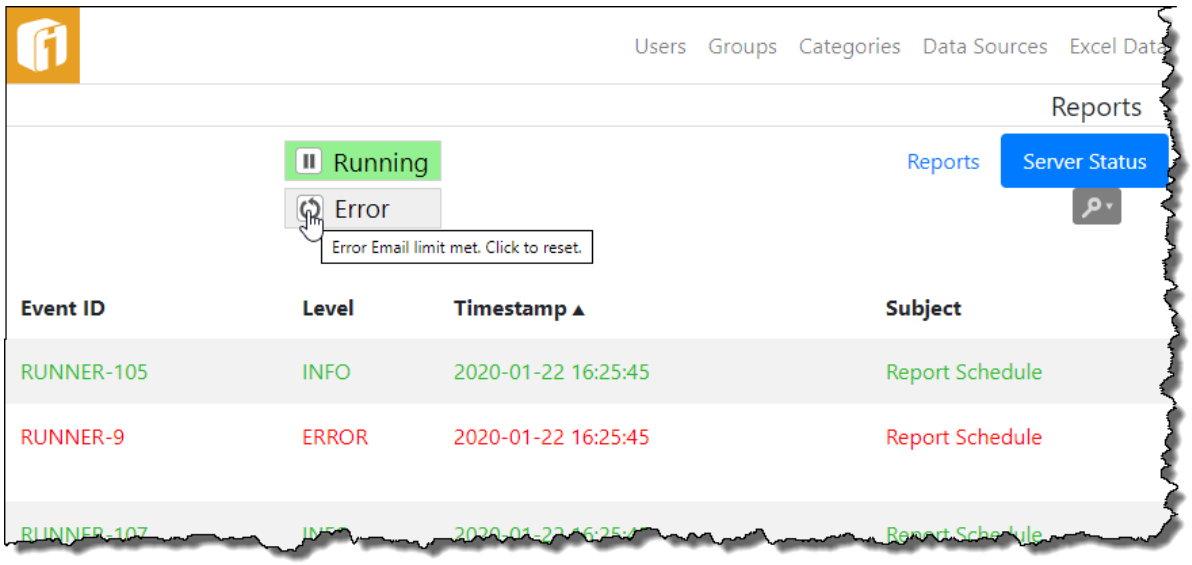
18.2.2.5 Email Events

Certain event types are designated as "email events". When an email event occurs, a notification email will be sent to the designated Reports Server administrators, provided that:

- The Reports Server is properly configured to send event notification emails.
- The level of the event (INFO, WARNING, or ERROR) is at or above the configured threshold at which the event notification emails are sent.

To determine whether or not an event in the list is an email event, hold the mouse cursor over its event ID until the tool tip appears. It will include the line "Email: true" for email events, and "Email: false" for non-email events.

When an ERROR qualified event occurs it will send the email and set the Email Limit button to "Error", with an "Error Email limit met. Click to reset." tooltip.

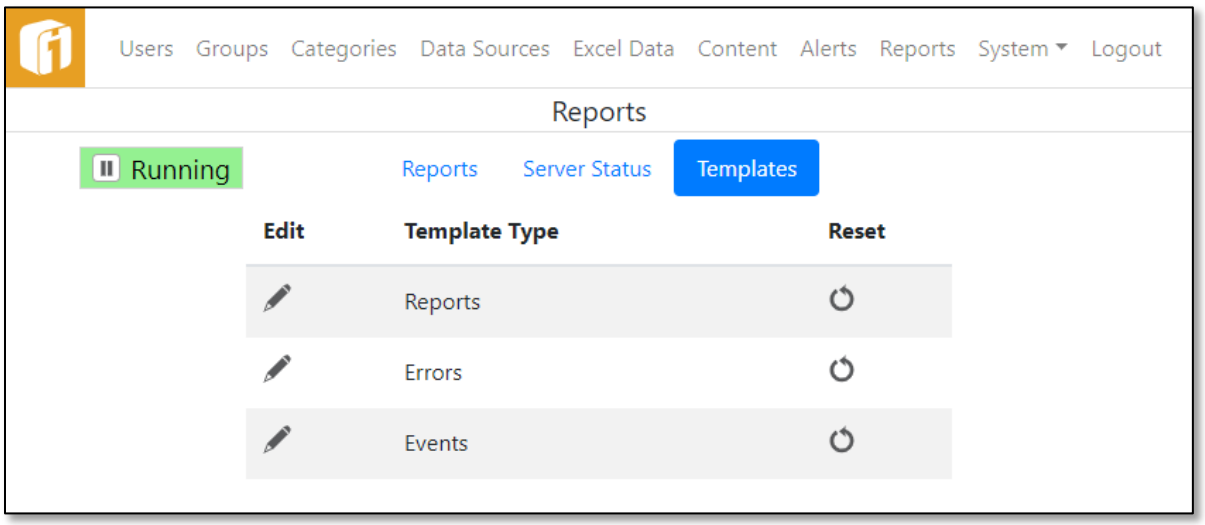


This will prevent the R Server administrators from continually receiving notifications for the same issue. To begin receiving ERROR emails, select the "Error" button to reset the limit counter. This will update the button to say "OK".

18.2.3 Templates

This is an optional step that provides a great deal of control over the information included in the bodies of notification emails messages. Using templates, notification emails can be sent in both HTML format (including images) and plain text. If left untouched, notifications will be sent as plain text and include only a minimal amount of default information.

Templates are managed through the Templates screen. To access the Templates screen, select Templates from Reports Administration.



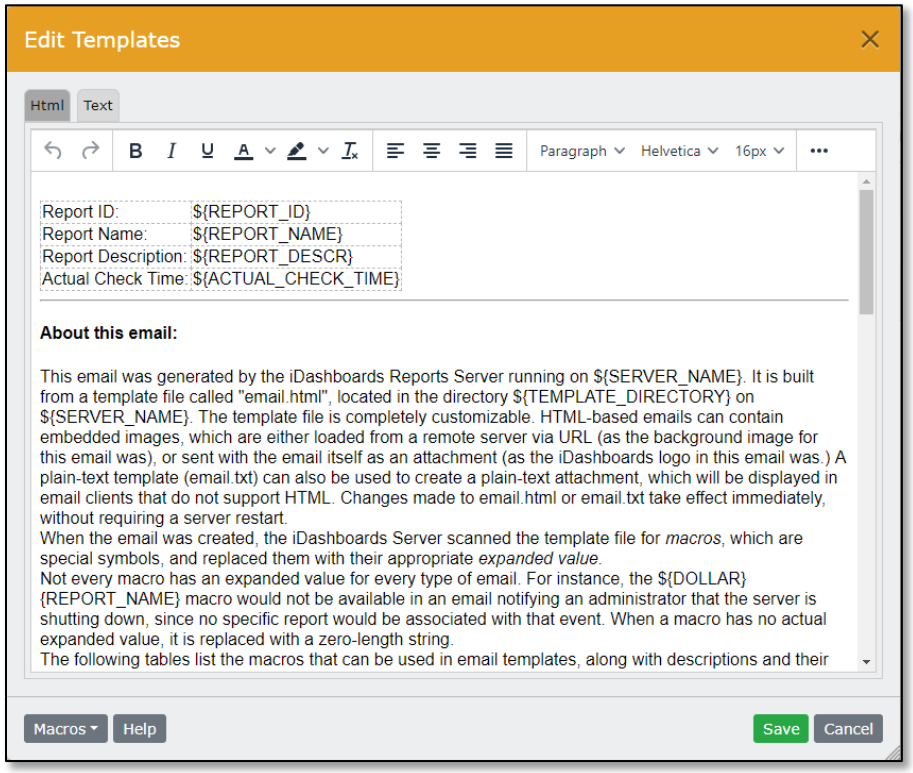
There are 3 Report Template Types:

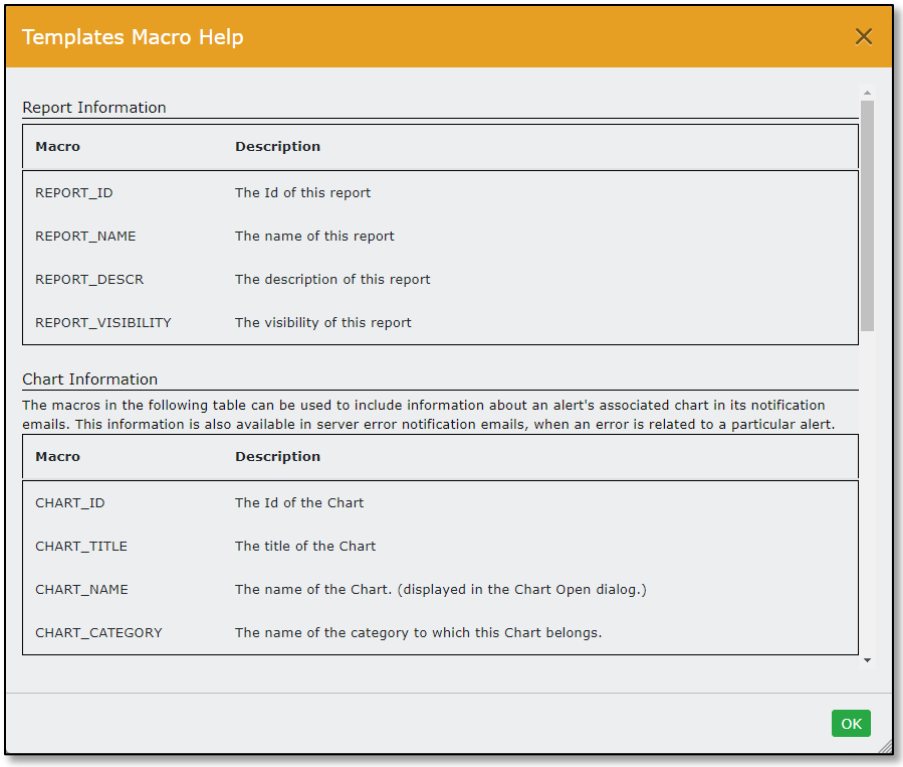
Type	Format	Usage
Reports	HTML/Text	Reports by Email
Errors	HTML/Text	Reports Server Error Notifications by Email.
Events	HTML/Text	Reports Server Event Notifications by Email.

To modify a Template, click its Edit icon ( ). The HTML panel provides a rich text editor for formatting and laying out an HTML emails, and the Text panel has a simple plain text box for plain text messages. The first time a Template is edited the system creates sample HTML and Text notifications, with examples of using the available alert macros. Macros can found and selected by using the Macros button. The Help button will also provide details about the template macros.

After the first save of a template, the template files are created in an iDashboards' application directory, identified by the ${TEMPLATE_DIRECTORY} macro.

Using a Template's Reset icon ( ) will regenerate the system created sample HTML and Text notifications for it, and also create the template files in ${TEMPLATE_DIRECTORY}.



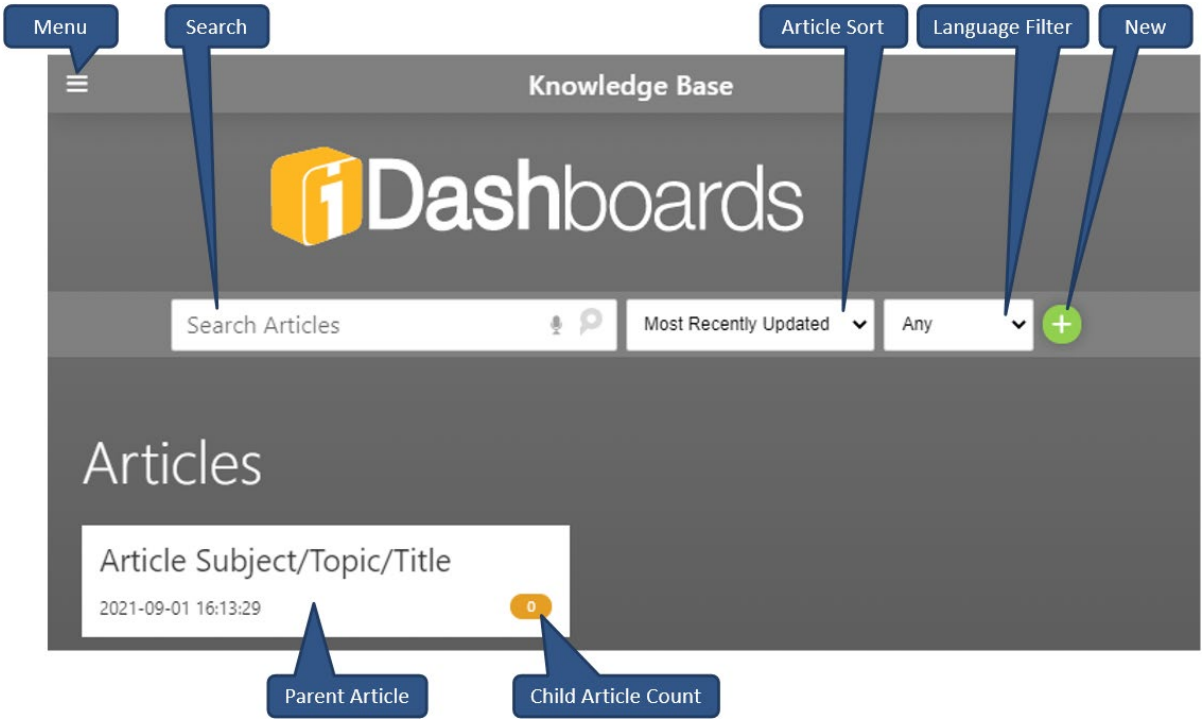

Templates Macro Help
Report Information
Macro
Description
REPORT_ID
The Id of this report
REPORT_NAME
The name of this report
REPORT_DESCR
The description of this report
REPORT_VISIBILITY
The visibility of this report
Chart Information
The macros in the following table can be used to include information about an alert's associated chart in its notification emails. This information is also available in server error notification emails, when an error is related to a particular alert.
Macro
Description
CHART_ID
The Id of the Chart
CHART_TITLE
The title of the Chart
CHART_NAME
The name of the Chart. (displayed in the Chart Open dialog.)
CHART_CATEGORY
The name of the category to which this Chart belongs.
OK

19. Knowledge Base

Note: The Knowledge Base feature must be enabled within the iDashboards license. The administrator has the option to disable the entire Knowledge Base feature.

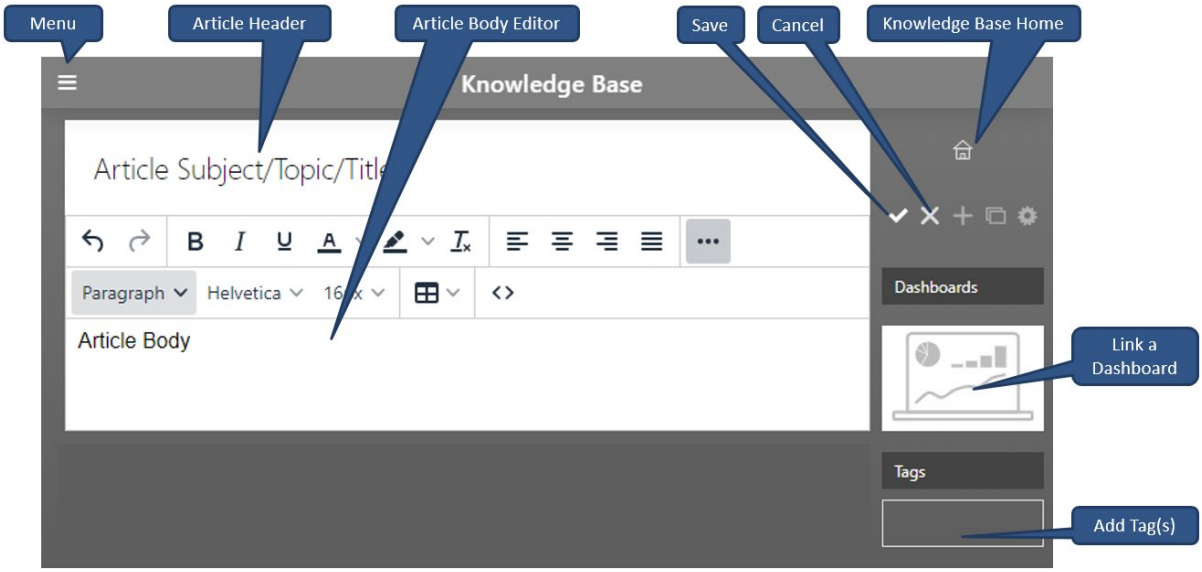
The iDashboard's Knowledge Base is an information repository used to facilitate the assembly of 'Articles'. These can be regarding general or specific content, like a topic, frequently asked questions, glossaries, instructions, etc.

19.1 Knowledge Base Home



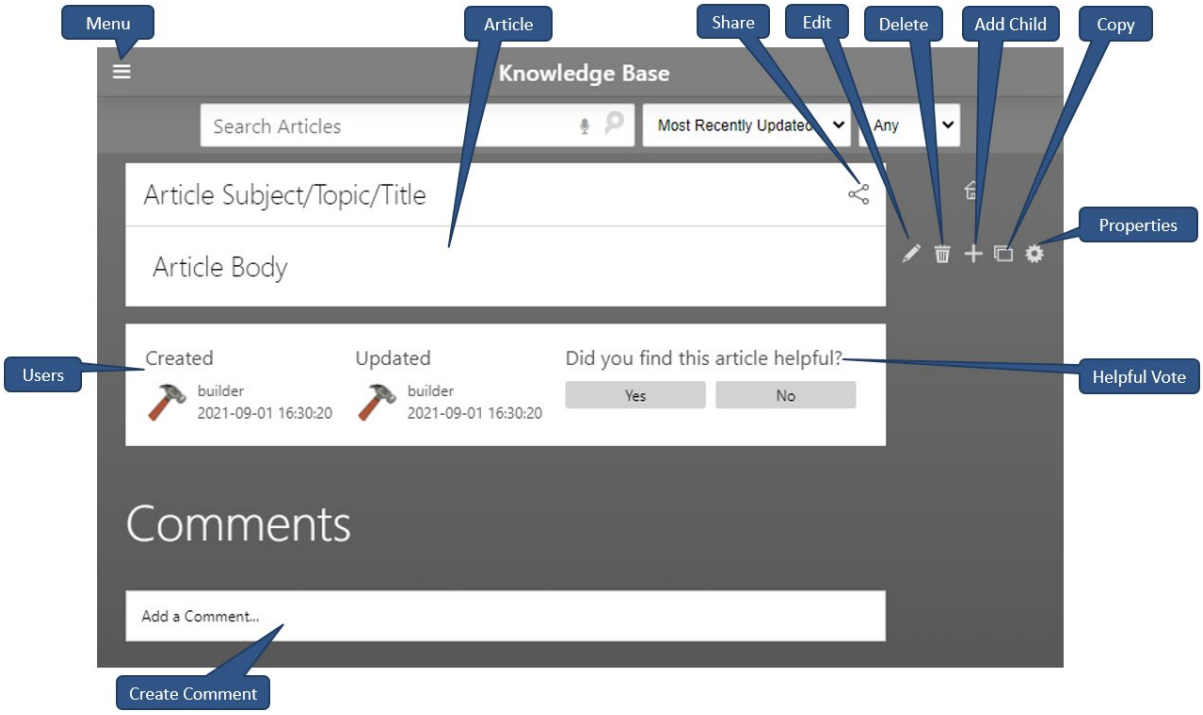
Articles are listed and created, using the 'New' button, from the Knowledge Base home. They can also be searched, sorted and filtered. Selecting an article will open it in view mode.

19.2 Create Article



Using the 'New' button, from the Knowledge Base home, begins an article. At this level they are considered the 'Parent'. One or more Dashboards can be attached to the article, along with Tags Once the article is saved, it will remain open in 'View' mode.

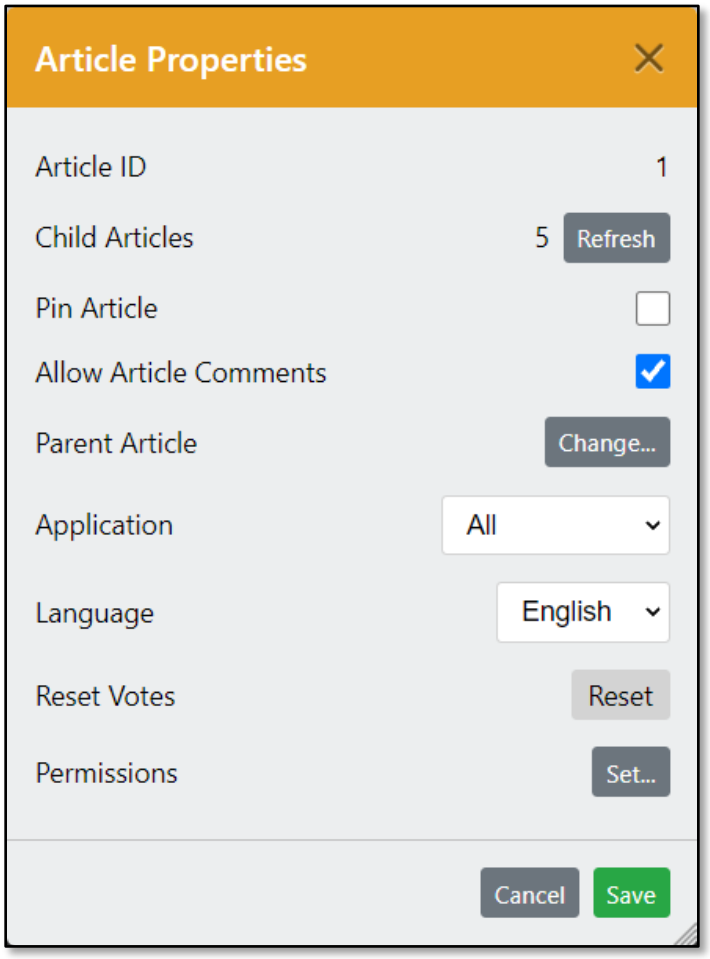
19.3 View Article



While viewing an article, along with editing and deleting it, the following tasks are available:

- Share – Provides a link to the article that can be copied or emailed.
- Add Child – Create and attached a sub article, referred to as a 'Child', to the main 'Parent' article. The same create article process as the 'Parent' is used.
- Copy – Creates a new article that is a copy of the current article. If the current article is a 'Child', the copy will also be a child of the same parent. 'Copy of', is added to the beginning of its name. This new article is in 'View' mode, so select 'Edit' button to make changes.
- Properties – Allows the viewing control of various settings, like commenting, changing the parent, and setting permissions (see Section 19.3.1 Properties).
- Helpful Vote – Results for all votes are calculated and used for the 'Most Helpful' / 'Lease Helpful' sorting.
- Comments – 'Add a Comment…' opens a simple editor, allowing a comment to be attached to the article. Existing comments are hidden by default and can be retrieve using the 'Show All Comments' button (see Section 19.3.2 Comments).

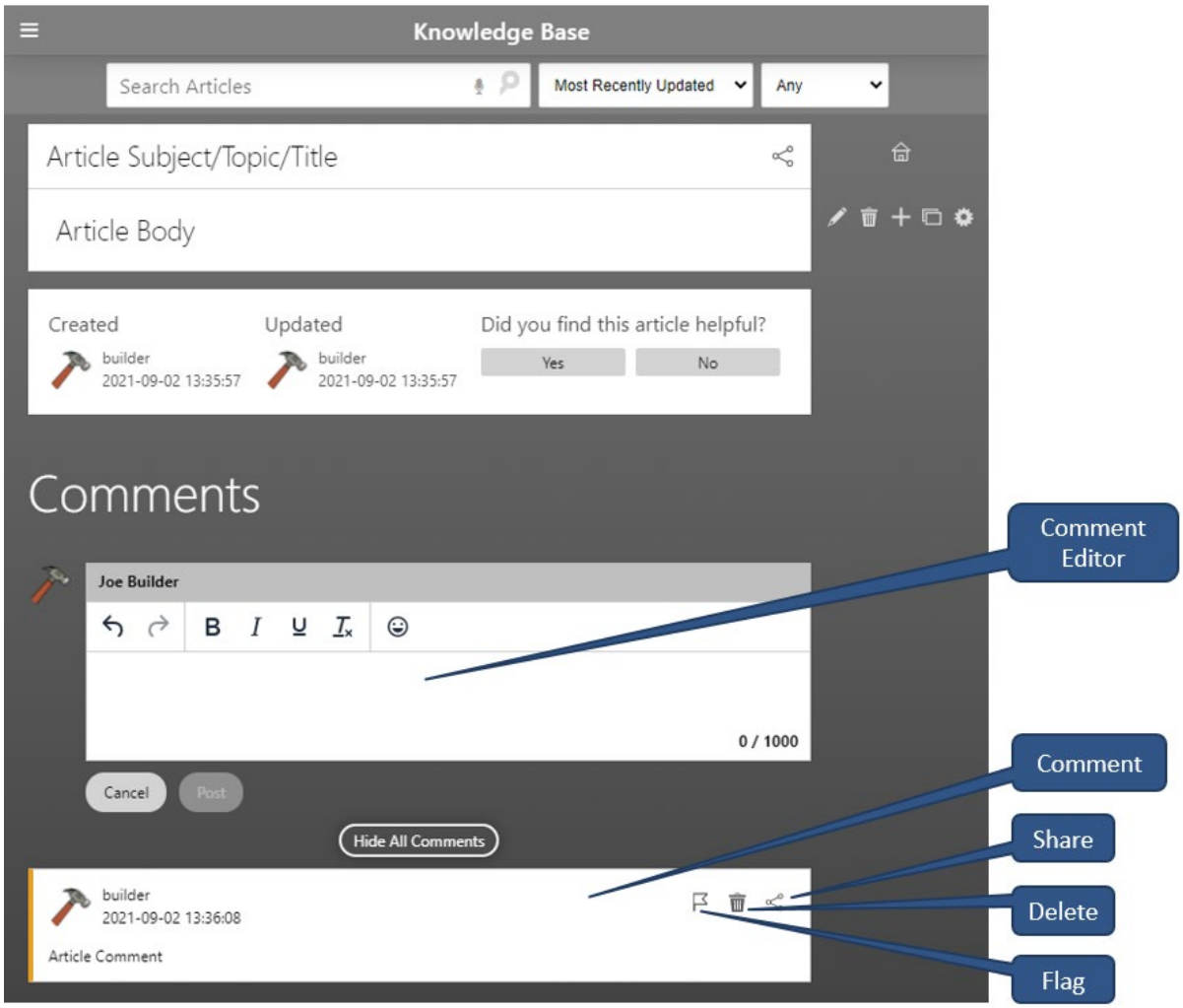
19.3.1 Article Properties



The components of Article Properties are:

- Article ID – The unique identifier for the article.
- Child Articles – The number of attached child articles.
- Pin Article – Force the article to appear toward the top of the list on the Home page.
- Allow Article Comments – Controls the ability to attached comments to articles.
- Parent Article – Move the child article to a different parent, or make it a parent article.
- Application – Associate the article will all (default), or one of the other installed applications.
- Language – Select a language, based on the installed language packs (see Section 13.6 Languages (Localization)).
- Reset Votes – Removes all helpful votes on the article.
- Permissions –
    - Require Authentication to View – This setting determines whether authentication is required to view articles. For controlling the system default see 13.2.12 Knowledge Base Settings.
    - Groups – The available privilege levels are "None", "Save" and "View". This is similar to category group privileges (see Section 9.1 Adding a Category).

19.3.2 Article Comments

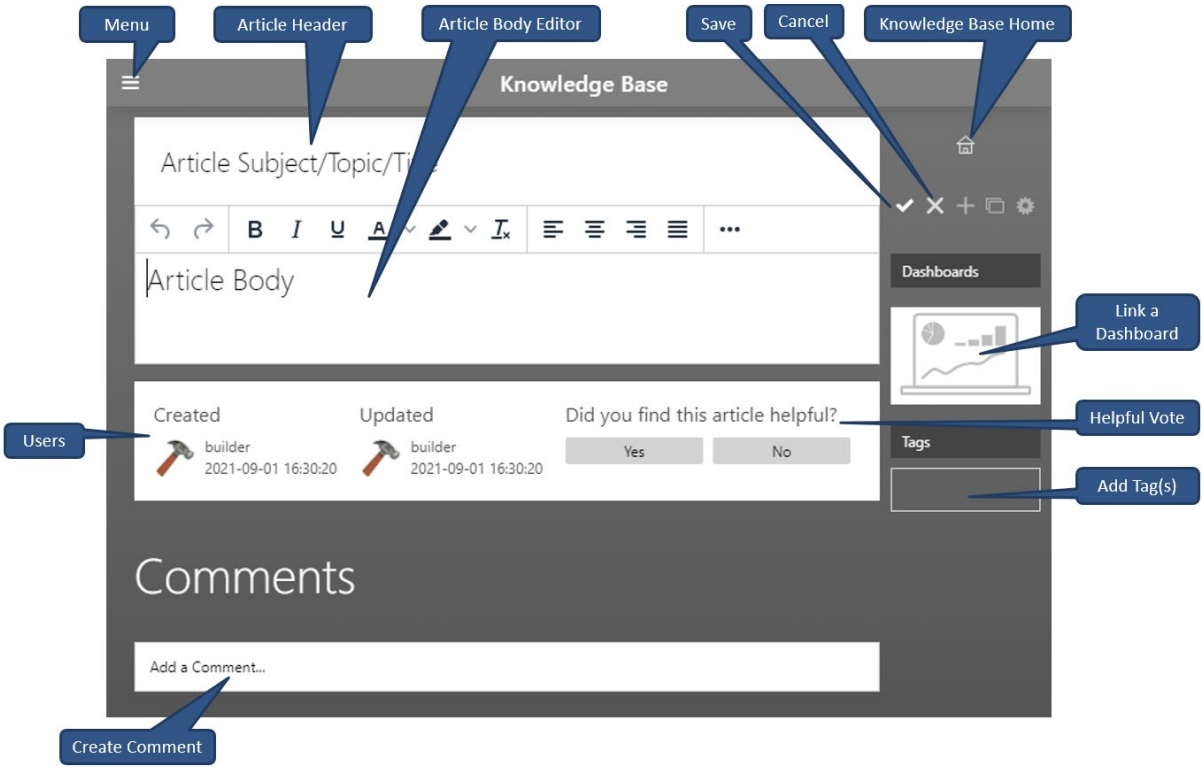


A comment can contain up to 1000 characters, with formatting and emoticons. After a comment is created, the comment editor will remain open, ready for another comment. If that is not needed, simply cancel or return to the Knowledge Base home.

These features are available:

- Toggling 'Show All Comments' / 'Hide All Comments' *(default)*.
- Flag a comment for moderation. This will remove the comment's context, but leaves the remaining features enabled. A moderator will review and remove the flag or delete the comment.
- Delete removes the comment from the article.
- Share provides a link to the article's comment that can be copied or emailed.

19.4 Edit Article



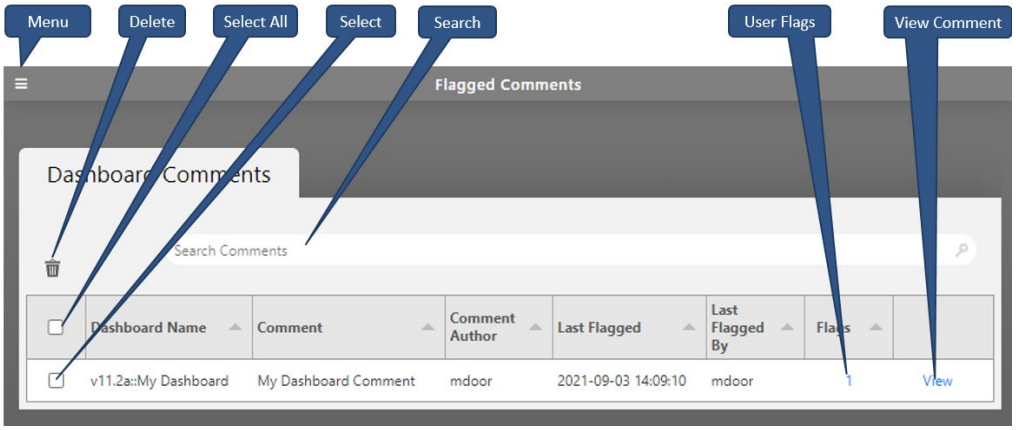
Editing an article, to modify its context, is similar to creating a new article. Other tasks that are available are the Helpful Vote and Comment creation. The created and last updated user's information and timestamps are also available.

20. Moderate Comments

A moderator maintains comments flagged for moderation. Moderators will not see the flag by viewing the comment itself, but only through this interface. Using the 'User Flag' link, a moderator will review and remove the flag from the comment, or delete the comment. The 'View Comment' link provides direct access to the full comment, as well as what it is attached to. Only users with a 'Builder' role and above can be moderators. All 'Admin' role users are automatically moderators; others need to be identified as moderators in their user settings (see Section 7.6 Comment Moderator Control).

20.1 Dashboard

Dashboard comments are added through its 'Comments Panel' (see Builder Manual's Section 6.17.6 Comment Panel).



20.2 Article

> *Note: The Knowledge Base feature must be enabled within the iDashboards license. The administrator has the option to disable the entire Knowledge Base feature.*

Article comments are added through the Knowledge Base (see Section 19.3.2 Article Comments).