



# DATAZEN

---

## Product Documentation

Version 3.0 | © 2015 Datazen Software Inc.

# Table of Contents

<b>INTRODUCTION</b>	<b>5</b>
<b>PRODUCT OVERVIEW</b>	<b>6</b>
PHILOSOPHY AND APPROACH	7
PRODUCT ELEMENTS	8
ARCHITECTURE OVERVIEW	9
<b>DATAZEN ENTERPRISE SERVER</b>	<b>10</b>
INTRODUCTION AND KEY CONCEPTS	11
SERVER COMPONENTS	12
SECURITY ARCHITECTURE	13
DATA ACQUISITION AND CACHING	15
SERVER SCALING	16
CACHING ARCHITECTURE	18
INSTALLATION GUIDE	19
<i>Installation Requirements</i>	20
<i>Installing Datazen Enterprise Server</i>	23
<i>Installing a Typical Distributed Deployment</i>	30
<i>Configuring Shared Caching with memcached</i>	32
<i>Configuring Shared Caching with Redis</i>	33
<i>Configuring Shared Caching with Azure Cache</i>	34
<i>Configuring ADFS authentication</i>	36
<i>Advanced Installation Scenarios</i>	39
<i>Security Best Practices</i>	40
INITIAL CONFIGURATION	44
<i>Authentication</i>	45
<i>Initial Users</i>	47
<i>Hub Configuration</i>	48
<i>Publishing Initial Content</i>	51
USING THE CONTROL PANEL	54
<i>Batch Creating Users</i>	55
<i>Managing KPIs</i>	56
<i>Managing Dashboards</i>	59
<i>Data Access Concepts</i>	61
<i>Data Connections</i>	65

<i>Managing Custom Maps</i> .....	70
<i>Managing Hub Users and User Groups</i> .....	71
<i>Managing Permissions</i> .....	75
<i>Managing Custom Branding</i> .....	77
<i>Managing Email Templates</i> .....	78
<i>Managing the Dashboard Runtime</i> .....	80
USING THE WEB VIEWER .....	81
<i>Accessing Dashboards and KPIs</i> .....	82
<i>Configuring Public Access</i> .....	84
<i>Configuring Integrated Windows Authentication</i> .....	85
<i>Embedding Datazen Content Into Custom Apps</i> .....	86
<i>Embedding Datazen Content Into SharePoint</i> .....	87
SERVER ADMINISTRATION .....	89
<i>Windows Services, IIS Sites, and Application Pools</i> .....	90
<i>Configuration Settings</i> .....	92
<i>Core Service Settings</i> .....	93
<i>Data Acquisition Service Settings</i> .....	99
<i>Datazen Rendering Service Settings</i> .....	101
<i>Control Panel Settings</i> .....	109
<i>Web API Configuration Settings</i> .....	110
<i>Server Logs</i> .....	112
<i>Backup and Restore</i> .....	113
<i>Repository Maintenance</i> .....	115
<b>CREATING AND PUBLISHING DASHBOARDS</b> .....	<b>116</b>
DASHBOARD DESIGNER .....	117
DASHBOARD RUNTIME .....	119
DATA .....	121
<i>Data Model</i> .....	122
<i>Preparing Dashboard Data</i> .....	124
<i>Working with Excel Data</i> .....	127
<i>Working with Live Data Sources</i> .....	130
<i>Working with Simulated Data</i> .....	131
CONFIGURING NAVIGATORS .....	132
CONFIGURING VISUALIZATIONS .....	136
PUBLISHING DASHBOARDS .....	143

ADVANCED TOPICS .....	145
<i>Parameterized Data Views and Load on Demand</i> .....	146
<i>Configuring Personalized Data Views</i> .....	148
<i>Implementing Row-Level Security for SSAS Data Views</i> .....	150
<i>Drill-throughs to Other Dashboards or Custom URLs</i> .....	155
<i>Hierarchical Selections</i> .....	157
<i>Cascading Selection Lists</i> .....	158
<i>Working with Custom Maps</i> .....	159
<b>EXTENDING DATAZEN</b> .....	<b>160</b>
IMPLEMENTING CUSTOM BRANDING .....	161
IMPLEMENTING CUSTOM MAPS .....	169
MANAGING DATA PROVIDERS .....	170

# Introduction

This material is designed to provide comprehensive coverage of the entire Datazen feature set, including detailed instructions on how to setup and configure Datazen Enterprise Server as well as create and publish dashboards and KPIs.

## Target audience

Product documentation is divided into two main categories:

1. [Datazen Enterprise Server](#): Written for system administrators who will be installing and configuring Datazen Enterprise Server, as well as power users who will be configuring BI hubs, connecting to live databases and creating data queries;
2. [Creating and Publishing Dashboards](#): Written for Business Intelligence professionals who will be creating dashboards based on previously created data queries.

## Prerequisites

This documentation is intended for those who already have a high-level understanding of the Datazen, obtained through the official website and the included product videos. System administrators should be proficient in configuring Windows Server, IIS, client-server networking and configuring connectivity to data sources.

# Product Overview

- [Philosophy and Approach](#)
- [Product Elements](#)
- [Architecture Overview](#)

# Philosophy & Approach

Datazen is designed to enable rapid development and publishing of business intelligence content in a way that delivers a premium user experience to any device. All product features are carefully conceived to simplify the process of connecting to enterprise data sources and securely delivering insight and analytics to any device type and form-factor. The following points outline our high-level philosophy and approach.

## Rapid Development





- **Use existing infrastructure.** Datazen doesn't impose changes of your existing technical infrastructure. The product integrates with your existing databases, data warehouses and the security infrastructure. It is designed to be an effective data analytics and BI presentation layer that works with the back end products that you already use today.
- **Build and publish rapidly.** The process of creating Datazen content - KPIs, dashboards, scorecards, and reports - is streamlined to balance simplicity and power. Enterprise data is discoverable and easily accessible through the data browsing UI of the Datazen Publisher app. Custom dashboard layouts are created in minutes. Full Datazen-based solutions can be deployed in a matter of days instead of months.
- **Create once, consume anywhere.** The Datazen Publisher app is the single point of dashboard creation and publishing. Once the data connections are made and dashboard layouts created, publishing a dashboard means immediate availability on all mobile platforms and device form-factors.

## Premium Experience on Any Device


- **Keep it simple.** Consuming BI content is easy and intuitive. End-users don't require any training to start using Datazen.
- **Prioritize mobile.** Datazen was designed for mobile scenarios first, then upscaled to traditional form-factors. As such, the product offers native apps for all mobile platforms, supports offline usage for disconnected scenarios, optimizes the entire UI for touch input and mobile screen form-factors, and ensures that data is secure in all mobile situations.
- **Make it beautiful.** Effective data analytics can also be made to look beautiful and not only empower, but delight users. Datazen's dashboard design model makes it impossible to create unattractive dashboards. Corporate branding integration takes this one step further and ensures an integrated, highly polished, branded experience throughout the entire product stack.


In summary, Datazen was created to address two major pain points in our industry: speed of deployment and quality of user experience, especially on mobile devices. The product enables quick publishing cycles and delivers timely insights to large audiences anytime, anywhere.

## Product Elements

Viewer Apps	Supported OS	Key Features
	Windows 8 Windows 8.1+	<ul style="list-style-type: none"> <li>• Connect to Datazen Server securely</li> <li>• Access published dashboards</li> <li>• Access published KPIs</li> <li>• Collaborate with team members in your BI hub</li> <li>• Store KPI data on the device for offline access</li> <li>• Store dashboard data on the device for offline access*</li> </ul>
	iOS 7+	
	Android 4.1+	
	Windows Phone 7.1 Windows Phone 8+	

\*Available only for Windows 8 in the current version.

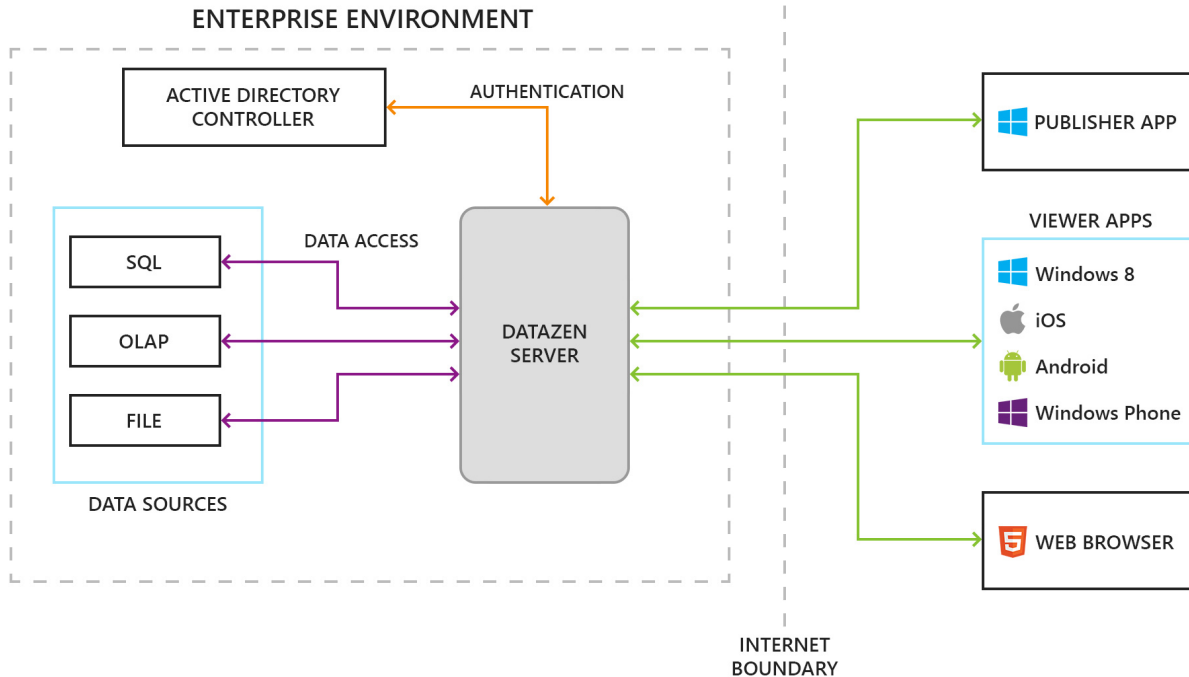
Publisher App	Supported OS	Key Features
	Windows 8 Windows 8.1+	<ul style="list-style-type: none"> <li>• Create and share dashboards from local Excel data</li> <li>• Create dashboards from cloud &amp; enterprise data sources and publish to any device</li> </ul>

Server	Supported OS	Key Features
	Windows Server 2008 R2 (64-bit) Windows Server 2012 (64-bit)	<ul style="list-style-type: none"> <li>• Connect to enterprise data sources</li> <li>• Integrate with Active Directory for user authentication</li> <li>• Publish dashboards for access by any device</li> <li>• Personalize data queries for each user</li> <li>• Integrate corporate branding</li> </ul>



# Architecture Overview

The following diagram shows a typical Datazen deployment:



Datazen Enterprise Server is typically deployed within an enterprise IT environment, integrated with Active Directory for [user authentication](#) and connected to internal data sources for [data access](#). Client applications (various viewer apps and the publisher app) communicate with Datazen Enterprise Server only; in other words *they don't access the data sources directly*. Client applications can access Datazen Server either from within the IT environment or outside the IT environment, if Datazen Server is [configured for external access](#).

For more information on product architecture please refer to the following sections:

- [Server Components](#)
- [Security Architecture](#)
- [Server Scaling](#)
- [Dashboard Runtime](#)

# Datazen Enterprise Server

- [Introduction and Key Concepts](#)
- [Server Components](#)
- [Security Architecture](#)
- [Server Scaling](#)
- [Installation Guide](#)
- [Initial Configuration](#)
- [Using the Control Panel](#)
- [Using the Web Viewer](#)
- [Server Maintenance](#)

# Introduction and Key Concepts

Datazen Enterprise Server is a collection of web applications and Windows services which:

- Act as a repository for storing and sharing dashboards and KPIs.
- Perform acquisition and caching of dashboard data.
- Secure dashboards and data for the organization.
- Provide an administrative Control Panel for managing and maintaining all aspects of the deployment.

## BI hubs

BI hubs are containers for grouping together an organization's users, dashboards, KPIs, data and security into a logical unit. A single Datazen Server instance can have many BI hubs provisioned. Users can be members of multiple hubs at the same time, potentially with different roles: hub owners, hub publishers or view-only users.

## Dashboard and KPI repository

One of the primary purposes of the Datazen Enterprise Server is to act as a central repository for dashboards and KPIs. Without Datazen Enterprise Server, dashboards can only be shared by e-mailing dashboard files and opening them with the Windows client on another machine. With Datazen Enterprise Server, dashboards and KPIs can be published to the server, organized and secured there, and retrieved at a later time by Datazen clients.

## Data acquisition and caching

Datazen dashboards use tabular data to power their rich visualizations. Without Datazen Enterprise Server, this data is limited to Microsoft Excel spreadsheets that are available to the author when the dashboard is created. With Datazen Enterprise Server data acquisition can be configured on the server and backed by various data providers. This data can then be added to visualizations by the dashboard author and is always up to date with the server, providing dashboard viewers with near real-time data capability.

## Dashboard and data security

Datazen Enterprise Server provides configurable membership and security features, allowing administrators to provide tight granularity over who can access dashboards and the data that drives them. User authentication can optionally be provided via Active Directory or ADFS.

# Server Components

Datazen Enterprise Server consists of the following software components:



## Core Service

- Windows service containing the Repository.
- Repository of users, KPIs, dashboards, data view definitions, custom maps, permissions, custom brand packages and cached data.
- Designed for fast read access by a large number of clients.

## Data Acquisition Service

- Windows service that periodically queries external data sources and caches results in the Core Service Repository.

## Rendering Service

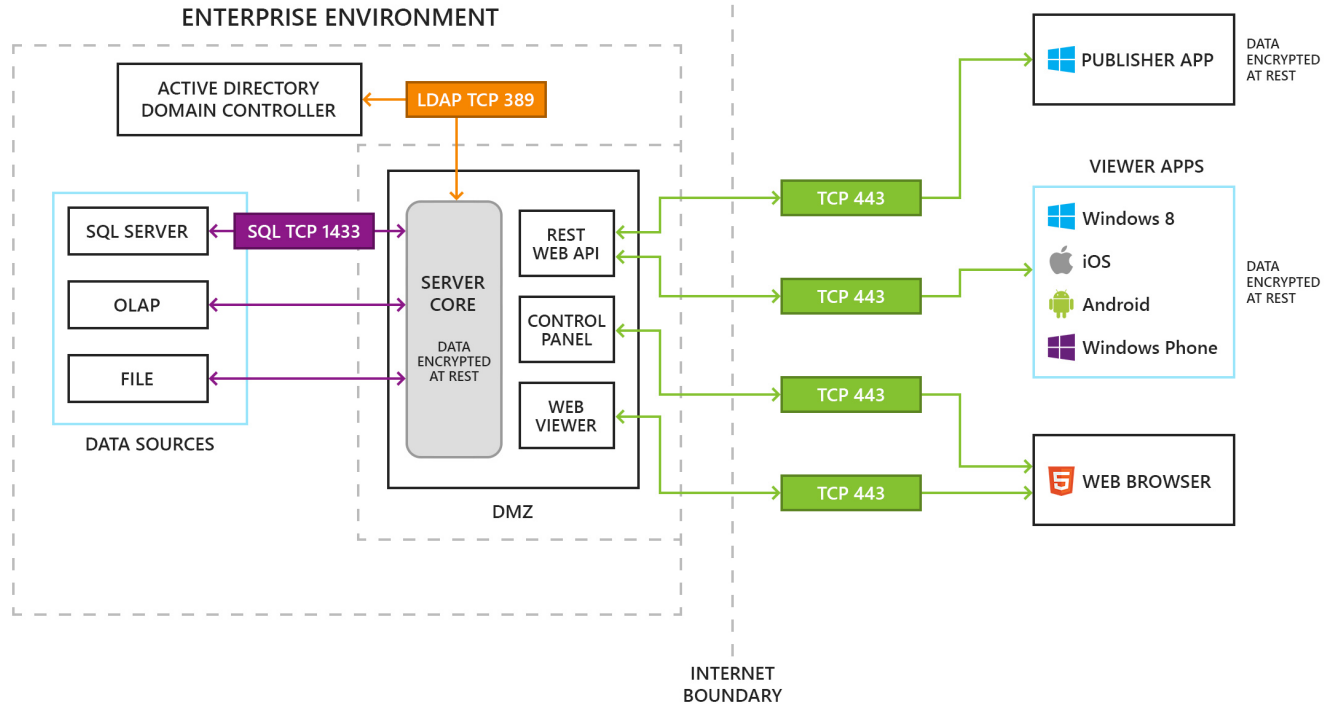
- Windows service responsible for processing and rendering dashboard thumbnails.

## Web Applications

- **Web API:** REST-based API that is the client-facing interface to all client applications.
- **Control Panel:** Application for browser-based server administration.
- **Viewer:** Web portal for browser-based access to dashboards and KPIs.

# Security Architecture

The following diagram describes the security architecture of a typical Datazen Enterprise Server deployment:



## Data encryption

Datazen implements data encryption at all levels of the product stack:

### Data is encrypted at rest when stored in the Datazen Enterprise Server Repository

The Datazen Enterprise Server Core Service Repository is built on [RavenDB](#), which supports full database encryption (AES-128).

### Data is encrypted for transit to Datazen Enterprise Server client devices

All Web Application client-server traffic can be configured to run over HTTPS. All client apps support HTTPS access to Datazen Enterprise Server

### Data is encrypted for transit between Datazen Enterprise Server Core Service and other Datazen back-end services

In a [distributed environment](#) all sensitive information (e.g. AD credentials, connection strings, etc.) is exchanged between the various Datazen services via a secure SSL channel.

### Data and user credentials are encrypted on each device

The Windows 8 client can cache server data for offline viewing. This data is encrypted using Windows DPAPI (User mode). Client apps store server credentials for the user. These are always encrypted and the method depends on the platform:

Client platform	Server credential encryption
Web Viewer	browser & OS dependent ("remember password" feature)

iOS	AES-128
Android	AES-128
Windows Phone	AES-128
Windows 8 / 8.1	Enhanced PasswordVault (Windows Credential Manager)

## Network considerations

It is recommended that Datazen Web Applications be deployed to a DMZ (allowing only port TCP/443 traffic inbound from the public network) while all other server components be deployed to the internal corporate network. See [Security Best Practices](#) for more information.

Server components of the Datazen Enterprise Server environment require access to Core Service on specific network ports. The following table outlines the Datazen Enterprise Server intra-service networking requirements for services that are [distributed](#):

Component	Required port	Description
Core Service	TCP/28952	Repository API
Core Service	TCP/28953	Management API

## User authentication

User authentication from Datazen client applications can be configured to utilize either Active Directory, an ADFS server, the Core Service Repository, or an external mechanism. See [Authentication](#) for more information.

## Managing access permissions

Granular access permissions can be provisioned for Datazen Enterprise Server KPIs, dashboards, data connections and activity feeds. See [Managing Permissions](#) for more information.

# Data Acquisition and Caching

While Datazen Enterprise Server enables access to many types of data, it is itself *not* a database. The underlying source data resides outside of the system, typically in a corporate database or data warehouse. Management of the source data is not handled by Datazen Enterprise Server.

[Data connections](#) to these external data sources are created and managed via the Datazen Enterprise Server [Control Panel](#). Various [data source types](#) are supported. Depending on the provider, a data connection may contain one or more data views. A data view corresponds to the tabular result set of a query (e.g. a SQL query, OData request, or worksheet within an Excel file on a network share). Each data view specifies the schedule for how often its data should be reacquired. The Data Acquisition Service periodically polls the Core Service for data views which require updating. It reacquires the data from the remote data source and caches the result in the Datazen Enterprise Server Repository. Real time views are never cached; they are acquired on demand by the Core Service.

Client applications access data via Datazen Enterprise Server and see all data views in a [standard, tabular way](#). Therefore client applications do not require access to the original data sources directly. Dashboard authors using the Datazen Publisher don't need to worry about data provider types nor how Datazen Enterprise Server acquires the data; they simply browse and discover available data views created and managed by administrators and BI hub owners.

## Real time data views

Data views can be specified as [real time](#). Real time data views are *not* refreshed by the Data Acquisition Service. Instead, they are retrieved from the original source by the Core Service every time they are requested.

## Cached data views

Unless specified as *real time*, data views are periodically reacquired by the Data Acquisition Service and cached in the Datazen Enterprise Server Repository. This ensures that data is readily available when it is needed by a dashboard or KPI. The update frequency is specified per data view.

## Personalized data views

Some data views can be [personalized](#). A personalized data view is one where a user's identity (*username*) is made available to the data view so that the data is acquired specifically for that Datazen user. All users who have [access to a personalized data view](#) (via the owning data connection) participate in personalization. Personalized data views may also be cached so that a unique data result is persisted for each user.

## Parameterized data views

Some data views can be [parameterized](#). Parameterized data views are configured to pass additional parameters to the underlying query. These parameters are then [connected to interactive elements on a dashboard](#). Parameterized data views are *real time* by definition, and therefore cannot be cached.

## Client side caching

Client applications have the ability to cache data views for offline dashboard viewing. This can be controlled per data view. When the *Allow Client Caching* checkbox is not set on the data view, client apps may not cache data on local devices.

# Server Component Scaling

The following [distributable](#) components of the Datazen Enterprise Server may be scaled out to more than one physical server machine:

- Data Acquisition Service
- Web Applications
- Shared cache instances

## Benefits

Any Datazen Enterprise Server component that is scaled beyond a single instance provides the immediate benefit of redundancy and reduced per-instance resource usage. Additionally, specific per-component benefits include:

Component	Benefit
Data Acquisition Service	Increased number of data views that can be updated per scheduled interval
Web Applications	Support a larger number of concurrent client connections
Shared cache instances	Increased number of items that may be cached

## Scaling configuration examples

### Scaling Web Applications with shared cache

This configuration supports an increased number of concurrently connected clients. Due to the utilization of shared cache the resource usage of the entire server environment is reduced.

1. Core Service, Data Acquisition Service and Rendering Service installed on a single physical server.
2. Multiple servers with Web Applications installed.
3. A single shared cache machine (implementing *memcached* for example).
4. A front-end load balancer is required to distribute client traffic (TCP/80 and/or TCP 443) to the Web Application instances.

### Scaling Web Applications and Data Acquisition Service independently, with shared cache

This configuration supports an increased number of concurrently connected clients and a greater number of scheduled data views that can be simultaneously acquired. Due to the utilization of shared cache the resource usage of the entire server environment is reduced.

1. Core Service and Rendering Service installed on a single physical server.
2. Multiple servers with Data Acquisition Service installed.
3. Multiple servers with Web Applications installed.
4. A single shared cache machine (implementing *memcached* for example).
5. A front-end load balancer is required to distribute client traffic (TCP/80 and/or TCP 443) to the Web Application instances.

## Other considerations



- Multiple instances of the Data Acquisition Service may be installed on separate machines. They work independently and divide the data acquisition workload accordingly.
- Multiple *memcached* or *Redis* caching instances may be installed and are then used in a load-balanced way.

For a step-by-step guide to setting up this fully distributed and scalable environment, see [Installing a Typical Distributed Deployment](#).

# Caching Architecture

A Datazen Enterprise Server deployment can significantly benefit from caching oft-accessed items. The primary benefit of enabling caching is to prevent the Core Service Repository from having to retrieve the same data items again and again if it has already successfully retrieved it once within the cache interval window, thus alleviating load on the Core Service.

Caching is utilized at the main entry points accessed by end-users: Web API and Web Viewer. It can be configured in standalone and distributed scenarios. In the latter case, a shared cache should be used (eg. memcached, Redis or Azure Cache).

## Web API and Web Viewer

Caching can be configured for the Web API application, the Web Viewer application, or both. Since these components directly interface with the Datazen Enterprise Server Repository and handle requests from desktop, mobile clients, and browsers, the benefits of caching are significant.

Some examples of items that are cached include:

- User manifests
- User authentication results
- Dashboard thumbnails
- Dashboard definitions
- Dashboard data

The Web API and Web Viewer applications can utilize an in-memory cache (for single instance or testing purposes), or a shared cache service (memcached, redis or Azure Cache). Please see [Web API Configuration](#) and [Web Viewer Configuration](#) for more information.

# Installation Guide

- [Installation Requirements](#)
- [Installing Datazen Enterprise Server](#)
- [Installing a Typical Distributed Deployment](#)
- [Configuring Caching with memcached](#)
- [Configuring Caching with Redis](#)
- [Configuring Caching with Azure Cache](#)
- [Configuring ADFS authentication](#)
- [Advanced Installation Scenarios](#)

# Installation Requirements

Datazen Enterprise Server can be deployed in standard Windows Server environments on physical machines, virtual machines or Azure VMs. This document details the minimum software and hardware requirements, as well as the recommended deployment configuration for common scenarios.

## Hardware requirements

The following configuration recommendations assume approximately 10% concurrent user load.

### Core Service

For server machines running Datazen Enterprise Server Core Service:

Recommended configuration:

Up to ~7500 users, minimal real time data access\*:

- CPU: 8 cores
- RAM: 16GB
- Available Disk Space: 250Gb or more

Up to ~10000 users, minimal real time data access\*:

- CPU: 10 cores
- RAM: 32GB
- Available Disk Space: 500Gb or more

\*increased real time data access will utilize greater CPU resources.

### Web Applications

For server machines running Datazen Enterprise Server Web Applications:

Up to ~650 users:

- CPU: 4 cores\*
- RAM: 16GB
- Available Disk Space: 5Gb

Up to ~1000 users:

- CPU: 8 cores\*
- RAM: 16GB
- Available Disk Space: 5Gb

Up to ~10000 users:

- CPU: 64 cores\*
- RAM: 16GB
- Available Disk Space: 5Gb

\*Web Application cores may be distributed across multiple machines.

## Software requirements

Datazen Enterprise Server can be installed on any of the following operating systems:

- Windows Server 2008 R2 64-bit with
  - .NET Framework 4.0
  - [.NET Framework 4.0 Update \(KB2468771\)](#)
- Windows Server 2012 64-bit (or later)

The *Web Applications* feature requires additional Windows features to be installed. The Datazen Enterprise Server [installer](#) will install and configure these features if they are not already configured on the target server:

For Windows Server 2008 R2

- Application Server
  - .NET Framework 3.5.1
- Internet Information Services
  - Web Server
  - Application Development
    - ASP.NET
    - .NET Extensibility
    - ISAPI Extensions
    - ISAPI Filters
  - Security
    - Basic Authentication
    - Windows Authentication
- Management Tools
  - IIS Management Console

For Windows Server 2012 (or later)

- Application Server
  - .NET Framework 4.5
- Web Server (IIS)
  - Web Server
    - Application Development
      - ASP.NET 4.5
      - .NET Extensibility 4.5
      - ISAPI Extensions
      - ISAPI Filters
    - Security
      - Basic Authentication
      - Windows Authentication
    - Management Tools

- IIS Management Console

Additionally, in IIS Manager:

- On the server node, *Feature Delegation*, *Handler Mappings* and *Modules* require Read/Write

## Deployment Configuration

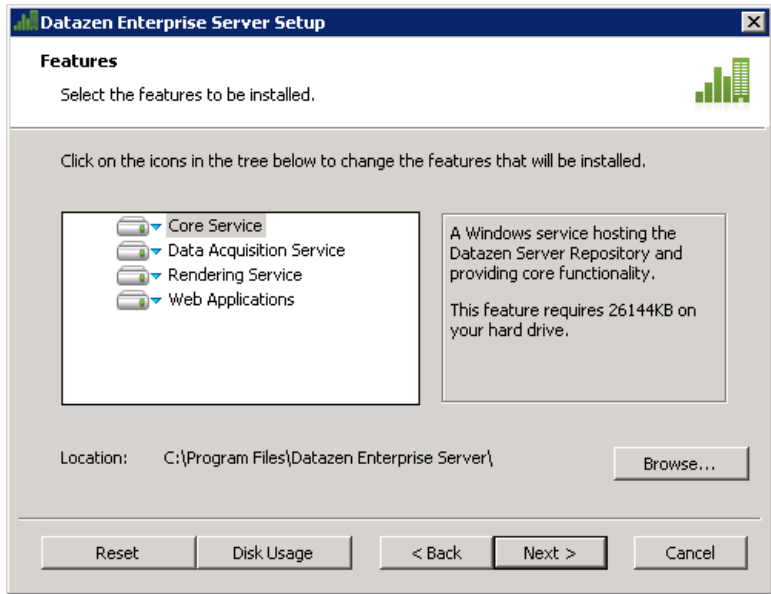
For development and testing deployments, a single server with all features installed should be sufficient. A single server configuration may also be used in production, supporting up to 500 registered users (assuming peak concurrent user load of 50 concurrent users). For performance, security and availability reasons a single-server deployment is not recommended.

For larger deployments, consider one of many possible scaling scenarios. Distributing the system across multiple servers is described in [Server Scaling](#).

# Installing Datazen Enterprise Server

The Datazen Enterprise Server installer contains the four major server features: [Core Service](#), [Data Acquisition Service](#), [Rendering Service](#), and [Web Applications](#).

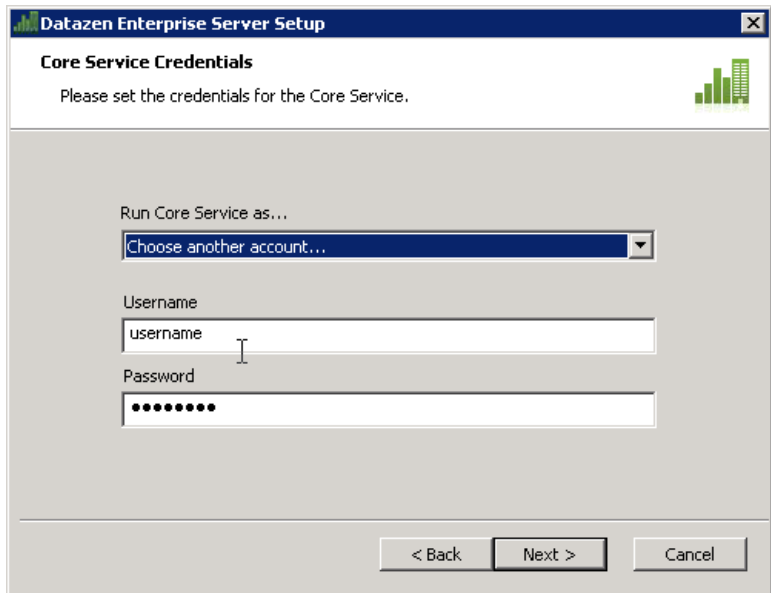
All of these features are required in order for the Datazen Enterprise Server to function. However, they do not need to be installed on the same machine. For various reasons it may be desirable to install the features together on a single-machine, or [distribute them among several machines](#).



## Core Service

The Datazen Enterprise Server Core Service contains the Repository and **must** be installed before any other features. This feature may **not** be distributed.

## Credentials



Use this dialog to specify the service account for Core Service. [It is recommended that a service account be provisioned for Core Service.](#)

If a service account is specified, the Datazen Enterprise Server installer will grant that account the following local rights and privileges:

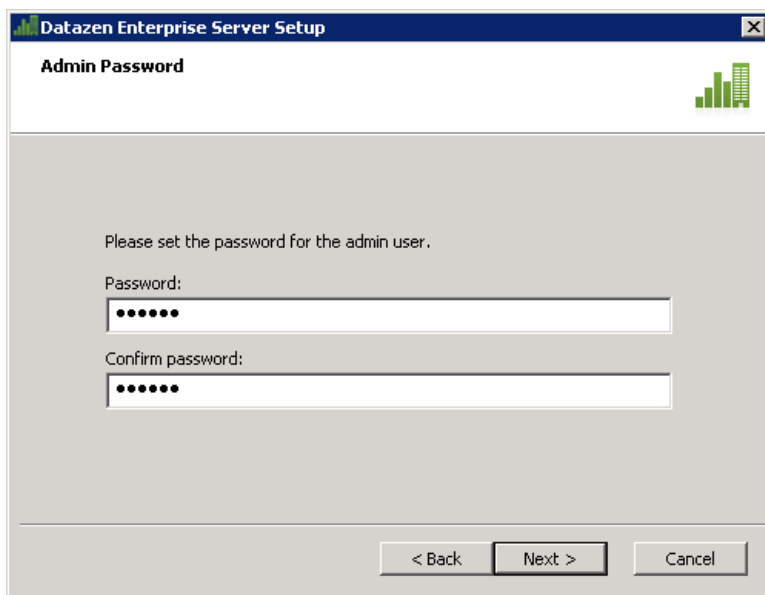
- Log on as a service
- Replace a process-level token.
- Back up files and directories.
- Log on locally.

If you are planning on using [Active Directory authentication](#) and this server is a domain member, the Core Service **must** be configured to run as a domain account. The installer will grant the specified user the *log on as a service* right on this machine.

**Changing the Core Service account:** If you change the service account for Core Service **after** Datazen Enterprise Server is installed (e.g. with Control Panel) the new account *must have the above rights and privileges configured*. Also, the new service account *must have URL reservations for the two Core Service endpoints*. To configure the URL reservations run the following commands (as administrator):

```
netsh http delete urlacl url=http://+:28952/
netsh http delete urlacl url=http://+:28953/
netsh http delete urlacl url=https://+:28953/
netsh http add urlacl url=http://+:28952/ user=[new service account username]
netsh http add urlacl url=https://+:28953/ user=[new service account username]
```

## Admin password



The screenshot shows a Windows-style dialog box titled "Datazen Enterprise Server Setup" with a sub-header "Admin Password". The main text says "Please set the password for the admin user." Below this are two password input fields: "Password:" and "Confirm password:", both containing six dots. At the bottom are three buttons: "< Back", "Next >", and "Cancel". A small green bar chart icon is in the top right corner of the dialog.

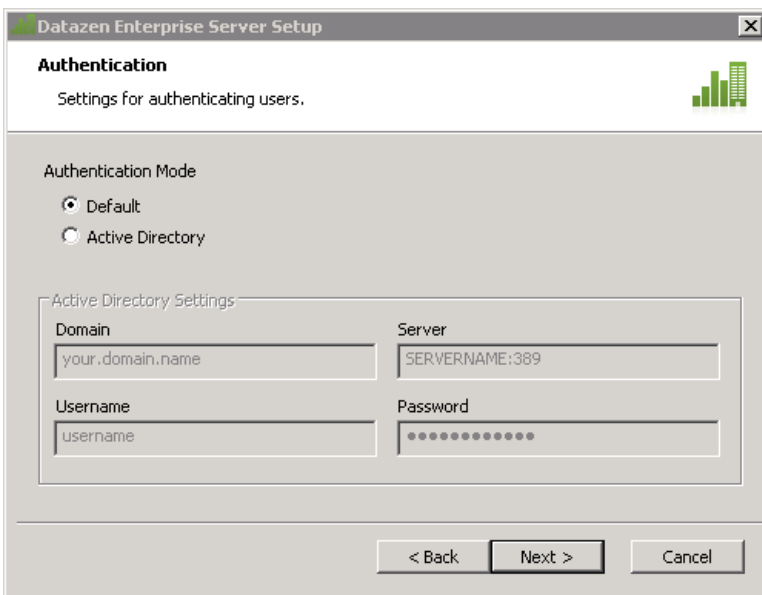
Enter the password for the Datazen Enterprise Server *admin* user. The password may be changed later via the [Control Panel](#).

## Authentication mode



Datazen Enterprise Server users can be authenticated in one the following modes:

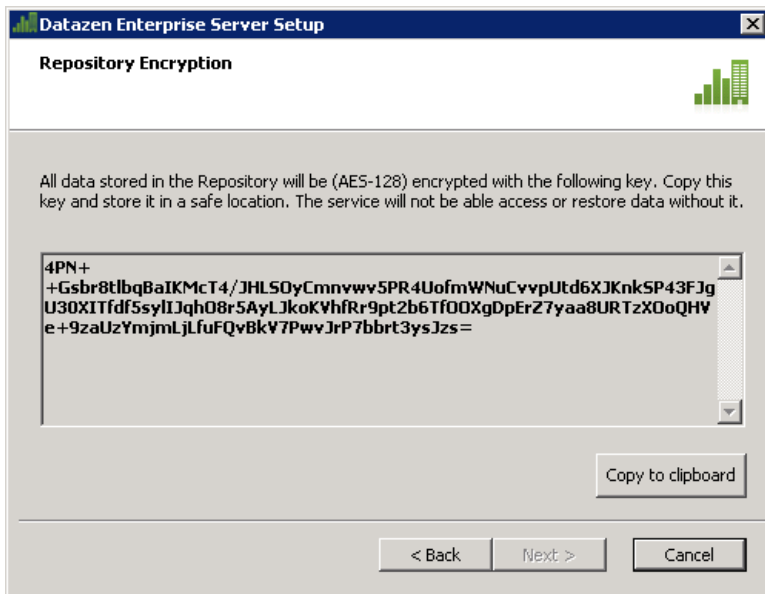
- *Default mode* - usernames and passwords are stored in the Core Service Repository.
- *Active Directory* - usernames are stored in the Core Service Repository. Passwords are authenticated against a local or remote Active Directory environment.
- *ADFS* - usernames are stored in the Core Service Repository. Users credentials are verified by an existing ADFS installation. **ADFS authentication is not configurable via the Datazen Enterprise Server installer.** If you are planning on using ADFS authentication, choose the *Default* option and configure ADFS after the installation is complete. See [Configuring ADFS](#) for details.
- *External* - usernames are stored in the Core Service Repository. Users credentials are verified by an existing external mechanism which sets a HTTP header or cookie value to the username. **external authentication is not configurable via the Datazen Enterprise Server installer.** If you are planning on using external authentication, choose the *Default* option and configure authentication after the installation is complete. See [Core Service Settings](#) for details.



When Active Directory is selected, the following options are available:

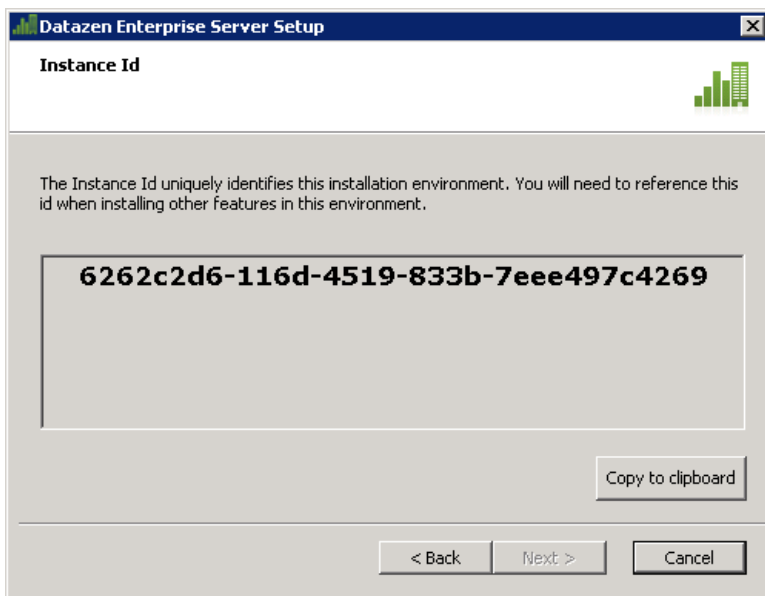
- *Domain* - If all of the users are in a single domain, enter the domain name here. Users can then be created by username only, instead of *username@domain.name*. If users are part of multiple domains, leave this blank.
- *Server* - If this machine is not a domain member, the Core Service will use LDAP to query Active Directory. Enter the LDAP server and port here. If this machine is a domain member, leave this blank.
- *Username* - If using LDAP, enter the username of the account that will perform Active Directory query. Otherwise, leave this blank and the query will be performed as the configured service user.
- *Password* - If using LDAP, enter the password of the account above. Otherwise, leave blank.

## Encryption Key



This is the Core Service Repository encryption key. This key is required in order to perform a [Core Repository restore](#). Ensure it is stored safely.

## Instance Id



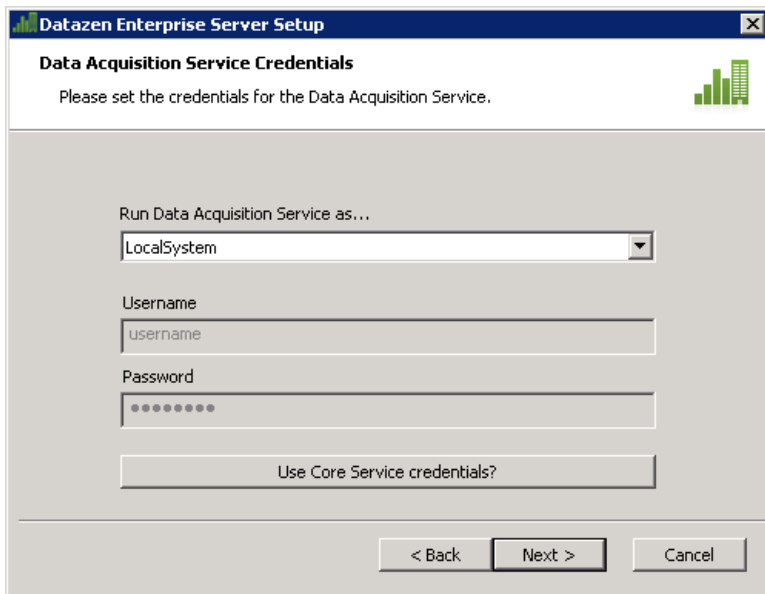
This is the Datazen Enterprise Server instance id. This key is required if [distributed components](#) will be used.

## Data Acquisition Service

The Data Acquisition Service is a Windows service that is responsible for [scheduled data acquisition](#).

This feature may be [distributed](#).

## Credentials



**Datazen Enterprise Server Setup**

**Data Acquisition Service Credentials**

Please set the credentials for the Data Acquisition Service.

Run Data Acquisition Service as...

LocalSystem

Username

username

Password

.....

Use Core Service credentials?

< Back   Next >   Cancel

Use this dialog to specify under which credentials the Data Acquisition Service will run.

Typically, this should be left as *LocalSystem* since the credentials for data acquisition are defined in a [Data Connection](#) connection string. If the data source requires a specific domain user context that cannot be defined in a connection string, those credentials can be entered here.

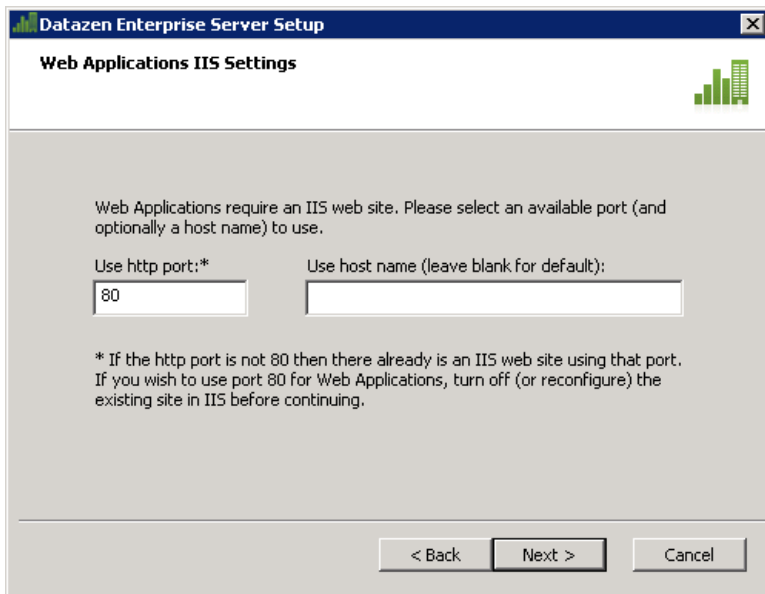
## Web Applications

This feature provides the following [Web Applications](#) to Datazen Enterprise Server client applications:

- *Web API* - all of the Datazen Enterprise Server client applications communicate directly with this service.
- [Control Panel](#) - the management portal used to manage users, dashboards, data, security and other aspects of the Datazen Enterprise Server installation.
- [Viewer](#) - a web portal for viewing published KPIs and dashboards in a web browser.

This feature may be [distributed](#).

## Internet Information Server



**Datazen Enterprise Server Setup**

**Web Applications IIS Settings**

Web Applications require an IIS web site. Please select an available port (and optionally a host name) to use.

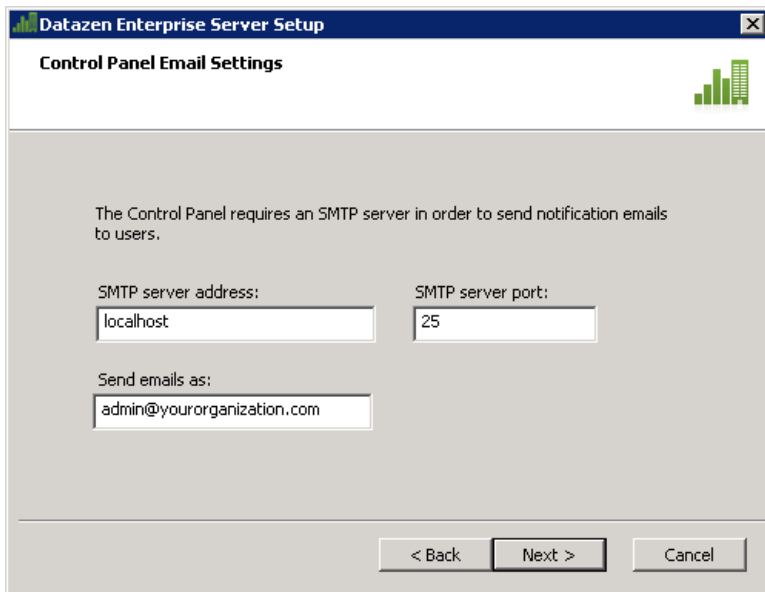
Use http port: \*  Use host name (leave blank for default):

\* If the http port is not 80 then there already is an IIS web site using that port. If you wish to use port 80 for Web Applications, turn off (or reconfigure) the existing site in IIS before continuing.

< Back Next > Cancel

Specify an available port (and optional hostname) on which Web Applications will run. A new website and several IIS application pools will be created. The website settings can also be configured anytime with the Windows IIS Manager (for enabling SSL security, for example).

## Control Panel Email



**Datazen Enterprise Server Setup**

**Control Panel Email Settings**

The Control Panel requires an SMTP server in order to send notification emails to users.

SMTP server address:  SMTP server port:

Send emails as:

< Back Next > Cancel

The [Control Panel](#) requires an SMTP server in order to send notification emails to Datazen Enterprise Server users.

## Rendering Service

The [Rendering Service](#) is a Windows service that is responsible for:

- Accepting requests from the [Viewer](#) and generating the HTML response.
- Generating thumbnails for published dashboards.

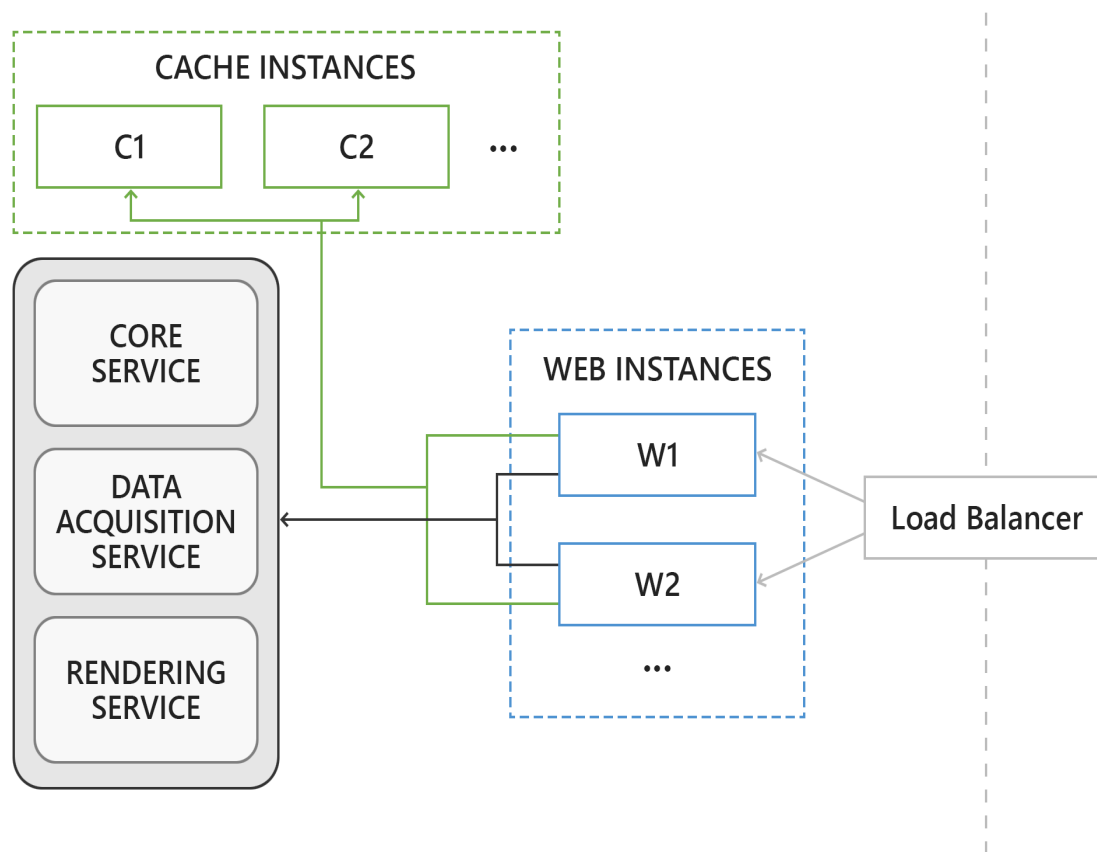
This feature may be [distributed](#).

This feature does not present any installation dialogs. For post-installation configuration of this service, please see the [Rendering Service configuration documentation](#).

# Installing a Typical Distributed Deployment

## Introduction

This document describes the process of installing and configuring a distributed and load-balanced deployment. The example provided utilizes [Core Service](#), [Data Acquisition Service](#), [Rendering Service](#) and [Web Applications](#) as separate, distributed and independent components.



Distributed and Load-balanced Architecture.

The example environment includes:

- 1 [Core Service](#), [Data Acquisition Service](#) and [Rendering Service](#) machine.
- 2 [Web Applications](#) machines (scalable independently)
- 2 Cache instances (scalable independently) for [shared caching](#)

## Server and network configuration

Once the machines have been provisioned ensure the Core Service is [port-visible to the Web Application machines](#). The load balancer should be configured to forward inbound requests (TCP/80 and/or TCP/443 for SSL) to the Web Applications machines. Web Applications are stateless, so the load balancing scheme should not implement affinity nor stickiness.

## Installing server components

1. On the Core Service machine, run the [installer](#) and select *Core Service*, *Data Acquisition Service* and *Rendering Service* as the features to be installed. Note the *instance id* as it will be needed for the next step.

2. On each Web Application machine, run the installer and select *Web Applications* as the sole feature to be installed. When prompted for the Core Service location, enter the ip address of the Core Service machine and the *instance id* recorded in step 1. When prompted for the Rendering Service location, enter the Core Service machine ip.

## Configuring shared caching

For more information see [Caching Architecture](#).

## Configuring distributed components

In a distributed environment, the [Control Panel](#) web application requires a shared, common [machineKey](#) across all Web Application instances. On each Web Application instance, edit the *controlpanel/web.config* file and ensure the machineKey property is set.

## Post-installation configuration

Proceed with [initial user and hub configuration](#). The Control Panel will be available on the load-balanced ip address, e.g. *http://load\_balanced\_ip\_address/cp*

# Configuring memcached

One or more [memcached](#) services may be configured for [shared caching](#) of Web Applications. The memcached service provisioning is not covered in this document, as there are several options depending on the server environment available. The memcached service is typically installed on a Linux, BSD or similar machine.

To configure memcached service for use:

1. Edit `/etc/memcached.conf` (or wherever the config is located).
2. Set the memory limit (in megabytes): e.g. `-m 1024`
3. Ensure that the service is listening on all interfaces by disabling the line: `-l 127.0.0.1`.

## Configuring Web Applications

Web API and Web Viewer instances need to be aware of memcached services. Ensure the following configuration is set in both the *webapi/web.config* and *viewer/web.config* configuration files:

```
<configuration>
  <appSettings>
    <add key="cache.enabled" value="true" />
    <!-- the default cache expiration in seconds -->
    <add key="cache.interval" value="60" />
    <add key="cache.type" value="memcached" />
    <add key="cache.memcached.protocol" value="binary" />
    <!-- the available memcached services as a comma delimited list -->
    <add key="cache.memcached.servers" value="10.0.0.7:11211,10.0.0.8:11211" />
  </appSettings>
</configuration>
```



# Configuring Redis

One or more [Redis](#) services may be configured for [shared caching](#) of Web Applications. The Redis service provisioning is not covered in this document, as there are several options depending on the server environment available. The Redis service is typically installed on a Linux, BSD or similar machine. [Azure Redis Cache](#) may also be utilized.

## Configuring Web Applications

Web API and Web Viewer instances need to be aware of Redis services. Ensure the following configuration is set in both the *webapi/web.config* and *viewer/web.config* configuration files:

```
<configuration>
  <appSettings>
    <add key="cache.enabled" value="true" />
    <!-- the default cache expiration in seconds -->
    <add key="cache.interval" value="60" />
    <add key="cache.type" value="redis" />
    <!-- optional Redis auth password -->
    <!-- for Azure Redis Cache this is the Azure key generated from the service portal -
->
    <add key="cache.redis.password" value="[password]" />
    <!-- redis service ssl enable/disable. Azure Redis Cache requires true, otherwise op
tional -->
    <add key="cache.redis.ssl" value="[true|false]" />
    <!-- the available Redis endpoints as a comma delimited list -->
    <!-- for Azure Cache Service, use the name of the cache endpoint -->
    <add key="cache.redis.endpoints" value="10.0.0.7:11211,10.0.0.8:11211" />
```

# Configuring Azure Cache

**IMPORTANT:** Datazen Server supports the *Azure Cache Service* only, not the *In Role Cache Service*.

The Azure Cache Service may be configured for [shared caching](#) of Web Applications.

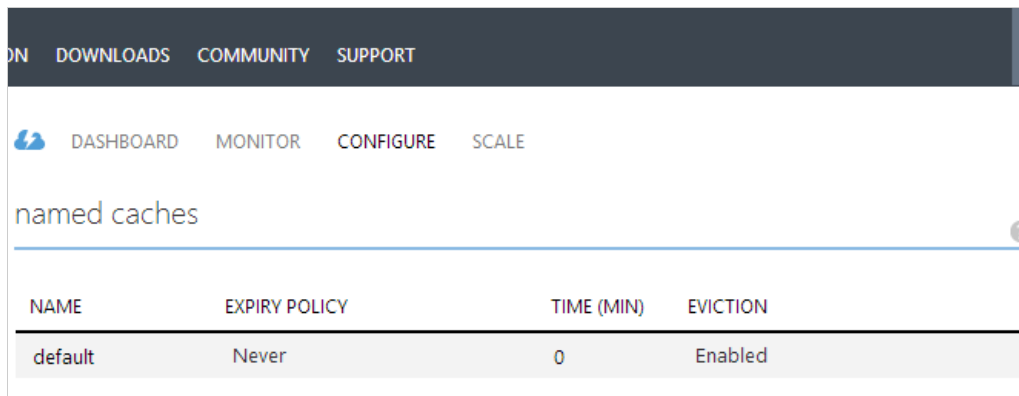
## Azure Cache Service

Ensure an Azure Cache Service [has been provisioned](#). The recommended options are:

- Cache Offering: *Basic (128MB to 1GB Memory)*
- Cache Memory: *1GB*

The Azure Cache Service settings should be set as follows:

- Expiry Policy: *Never*
- Time (Min): *0*



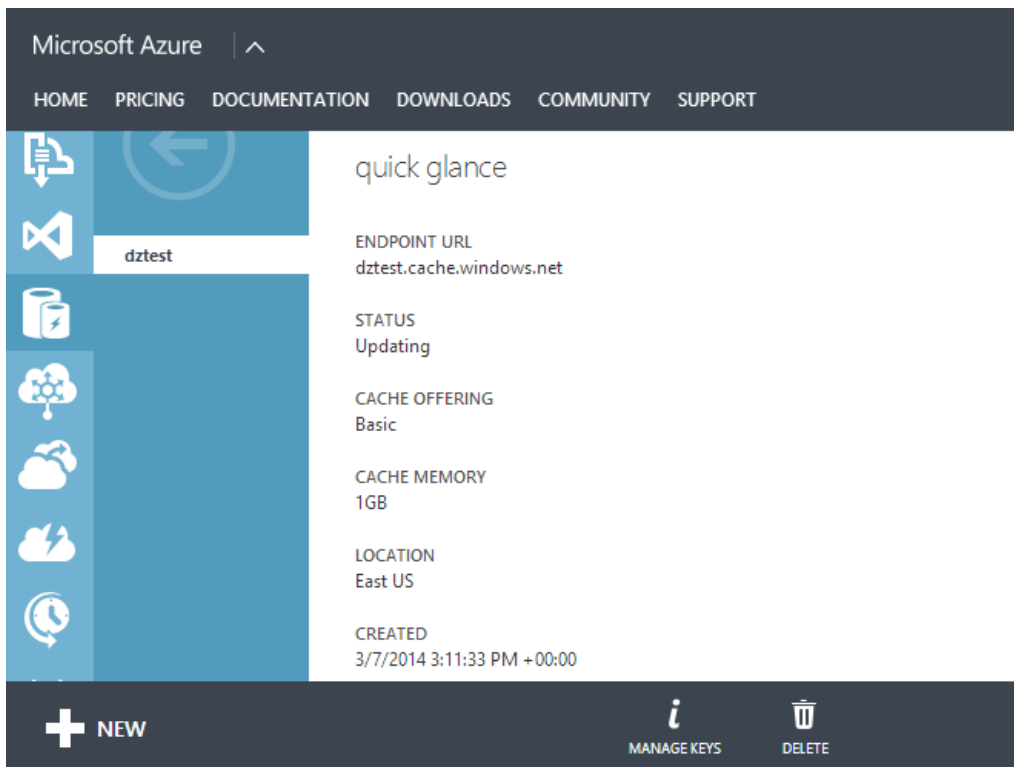
ON	DOWNLOADS	COMMUNITY	SUPPORT
DASHBOARD	MONITOR	CONFIGURE	SCALE
named caches			
NAME	EXPIRY POLICY	TIME (MIN)	EVICTON
default	Never	0	Enabled

## Configuring Web Applications

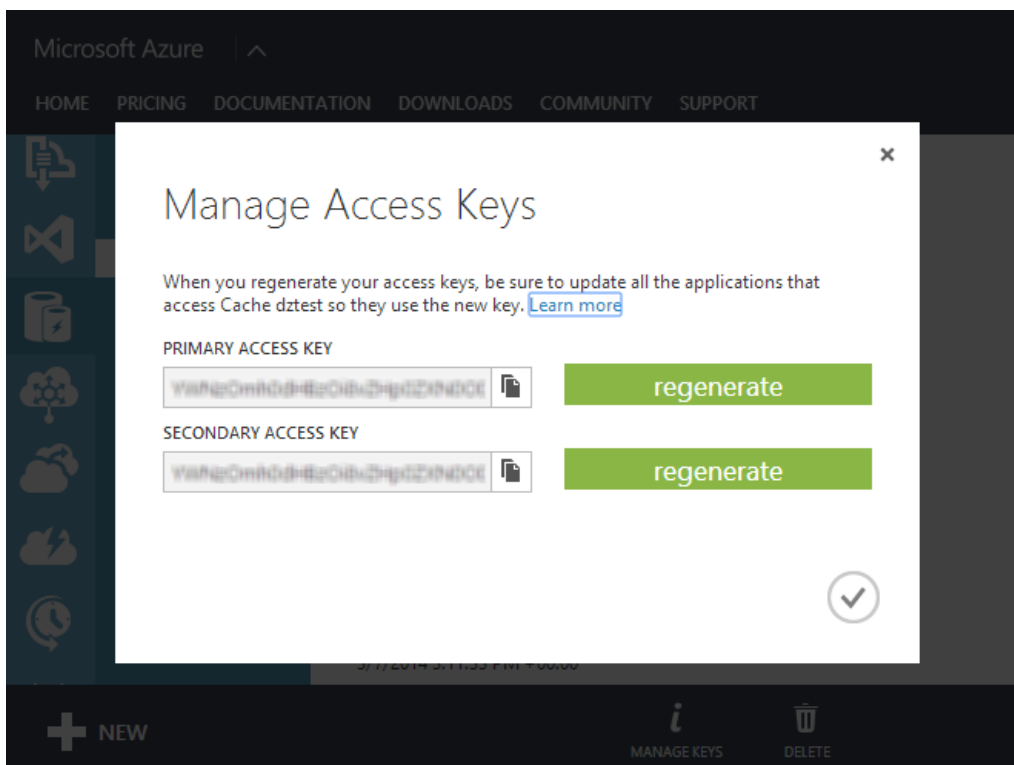
Web API and Web Viewer instances need to be aware of the Azure Cache Service. Ensure the following configuration is set in both the *webapi/web.config* and *viewer/web.config* configuration files:

```
<configuration>
  <appSettings>
    <add key="cache.enabled" value="true" />
    <!-- the default cache expiration in seconds -->
    <add key="cache.interval" value="60" />
    <add key="cache.type" value="azure" />
    <add key="cache.azure.cachename" value="default" />
    <add key="cache.azure.endpoint" value="{endpoint url}" />
    <add key="cache.azure.authtoken" value="{access key}" />
  </appSettings>
</configuration>
```

Replace the {endpoint url} with the *Endpoint URL* visible on the *quick glance* section of the Azure Management Console dashboard screen:



The {access key} can be obtained by clicking the *Manage Keys* button at the bottom of the Azure Management Console dashboard screen:



# Configuring ADFS authentication

ADFS authentication must be configured in the following locations:

- Datazen Control Panel
- Datazen Core Service
- ADFS server

## Preparation

**Note:** It is assumed that server administrators are familiar with [ADFS terminology](#) before attempting to configure ADFS authentication.

The following items are required for configuring ADFS authentication:

- The publicly available url of the ADFS server (via https).
- The publicly available url of the Datazen Web Api (via https).
- The ADFS audience uri.
- The STS token-signing certificate.
- The Relying Party certificate (including the private key).

## IMPORTANT NOTES ABOUT CERTIFICATES

- Datazen does not install or manage your ADFS certificates. Use the Microsoft Management Console's Certificate Management snap-in.
- The STS and RP certificates must already be installed on *all instances* of the Web Api, Web Viewer and Control Panel machine(s) before proceeding with ADFS configuration.
- Certificates must be installed in the *Local Computer Personal* certificate store. If a certificate is self-signed, it must also be installed in the *Local Computer Trusted Root Certification Authorities* store.
- The RP certificate must include the private key, and the following IIS Application Pool identities must have at least **read** access to the private key: *Datazen.Server.WebApi*, *Datazen.Server.ControlPanel*, and *Datazen.Server.WebViewer*. Use the Microsoft Management Console's Certificate Management snap-in to manage private key access.

## ADFS Settings

The settings for ADFS are available in the Datazen Control Panel, *Authentication* menu item:

Active Directory

**ADFS**

External

### Settings

ADFS server root:

Audience URI:

RP certificate:

...

ADFS token-signing certificate:

...

**Apply Changes**

Enter the following four items:

- *ADFS server root*: The publicly available url of the ADFS server. (e.g. `https://adfs.company.com`)
- *Audience URI*: The ADFS audience uri. (e.g. `http://my.application.com/adfs/services/trust`)
- *RP certificate*: Choose the previously installed RP certificate.
- *ADFS token-signing certificate*: Choose the previously installed STS certificate.

Once configured, click *Apply Changes*.

## Datazen Core Service Settings

In the Datazen Core Service configuration file, the `authtype` key's value must be set to `adfs`:

Location:

```
<configuration>
  <appSettings>
    <add key="authtype" value="adfs" />
```

This action must be performed when the Core Service is *stopped*. The authentication mode will take effect when the Core Service is started.

## ADFS server configuration

The Datazen Web Api exposes an endpoint containing the *federation metadata* of the configured Datazen ADFS settings:

`/api3/federatedmetadata.xml`

When creating the Relying Party Trust in ADFS, you can direct the wizard to access this endpoint directly:

The screenshot shows the 'Add Relying Party Trust Wizard' window. The title bar is blue with the text 'Add Relying Party Trust Wizard' and a close button. The window is divided into two main sections. On the left is a 'Steps' pane with a list of steps: 'Welcome' (green dot), 'Select Data Source' (green dot and highlighted), 'Configure Multi-factor Authentication Now?' (blue dot), 'Choose Issuance Authorization Rules' (blue dot), 'Ready to Add Trust' (blue dot), and 'Finish' (blue dot). The main area on the right is titled 'Select Data Source' and contains the following text: 'Select an option that this wizard will use to obtain data about this relying party:'. There are three radio button options: 1. 'Import data about the relying party published online or on a local network' (selected). Below this is the text: 'Use this option to import the necessary data and certificates from a relying party organization that publishes its federation metadata online or on a local network.' followed by a text box for 'Federation metadata address (host name or URL):' containing 'https://my.server.com/api3/federatedmetadata.xml'. Below the text box is an example: 'Example: fs.contoso.com or https://www.contoso.com/app'. 2. 'Import data about the relying party from a file'. Below this is the text: 'Use this option to import the necessary data and certificates from a relying party organization that has exported its federation metadata to a file. Ensure that this file is from a trusted source. This wizard will not validate the source of the file.' followed by a text box for 'Federation metadata file location:' and a 'Browse...' button. 3. 'Enter data about the relying party manually'. Below this is the text: 'Use this option to manually input the necessary data about this relying party organization.' At the bottom right of the main area are three buttons: '< Previous', 'Next >', and 'Cancel'.

Ensure the Relying Party Trust identifier matches the *audience uri*.

## Advanced customization

The automatically generated federation metadata may be copied, edited and imported directly into the ADFS server if further customization is required.

## Claim rules

Datazen ADFS authentication requires that the ADFS server return a *UPN* claim in the successful authentication response. The value of this claim must be the username of the authenticated user, and it must match the username of a registered Datazen user.

# Advanced Installation Scenarios

## Configuring the installer from the command line

The Datazen Enterprise Server installation executable supports command line parameters. These may be optionally set to force property values into the installer and should be used in the following scenarios:

- When Datazen Enterprise Server components exist but were not originally deployed using the Datazen Enterprise Server installation executable (e.g. components were previously moved from another machine post-install).
- When no previous Datazen Enterprise Server installation exists but a Core Service Repository backup will be [restored](#) post-install (i.e. disaster recovery).

In the above scenarios, the installer does not know about the previous (or planned) Datazen Enterprise Server installation settings. These settings may be injected into the installer via the following command line parameters:

### Instance Id

The *instance id* is randomly generated by the installer if [Core Service](#) is to be installed and no previous *instance id* is detected. In the above scenarios this value will not match the *instance id* of the running Datazen Enterprise Server environment. To force a specific *instance id* set the following command line option:

```
DATAZEN_INSTANCE_ID=[instanceid]
```

### Encryption Key

The *encryption key* is randomly generated by the installer if [Core Service](#) is to be installed and no previous *encryption key* is detected. In the above scenarios this value will not match the *encryption key* of the running Datazen Enterprise Server environment. To force a specific *encryption key* set the following command line option:

```
DATAZEN_ENCRYPTION_KEY=[encryption key]
```

**IMPORTANT:** If the target Datazen Enterprise Server environment does not use encryption (such as a pre-2.6 Core Service Repository) then encryption must be disabled on this installation via the following command line option:

```
DONT_USE_ENCRYPTION=true
```

# Security Best Practices

This document contains a list of recommendations for improving the security of your Datazen Enterprise Server environment. While following these recommendations does not guarantee freedom from security issues, these recommendations can significantly reduce your risk.

## Installation and Configuration

- Ensure the host machine is configured securely.

Many organizations have security policy in place to improve the security of servers in their environment. At a minimum, you should ensure that the server has all security updates applied, has up to date anti-virus software and that the local firewall is configured properly.

- Install the latest Datazen service packs.

The latest version of Datazen includes all the latest updates and should always be used.

- Install Web Applications and Core Service on separate machines.

By installing Datazen Enterprise Server Web Applications and Core Service [on separate machines](#), the ability to compromise data residing in Core Service via an attack originating from the web is significantly reduced. Only install the required components on each machine, and consider installing Web Applications in the DMZ and other Datazen components in the internal network.

- Secure the instance id and encryption key.

During the [initial installation](#) of Core Service, a random instance id and encryption key are generated which are unique to your Datazen Enterprise Server environment. Secure these items carefully, as they could be maliciously used to gain access to various Datazen Enterprise Server services and/or decrypt the Core Service repository.

## Core Service

- Do not run Core Service as LocalSystem.

During the [initial installation](#) of Core Service you are prompted for the credentials to run the service as. Choose a local service account, and not LocalSystem. The service account requires the following rights and privileges:

- Modify files in the Datazen Enterprise Server installation target folder.
- Log on as a service
- Replace a process-level token.
- Back up files and directories.
- Log on locally.
- Be a member of Performance Log Users group.

The Datazen Enterprise Server installer will configure the required security privileges for the chosen account. LocalSystem has privileges beyond what is required for Core Service operation, and can greatly increase risk should this service be compromised.

- Do not disable the account lockout feature.



The Datazen Enterprise Server [account lockout feature](#) restricts the number of authentication failures allowed before the account is administratively locked. If this feature is disabled the risk of a brute-force password attack is exposed.

- Only allow inbound traffic from other Datazen Enterprise Server services.

Core Service is exclusively utilized by [other Datazen Enterprise Server services and Web Applications](#). Consider setting up a firewall security policy that only allows inbound traffic from addresses/ports running other Datazen Enterprise Server components. The default network ports required by Core Service are: TCP/28952 and TCP/28953.

- Provision an encrypted tunnel.

Although much of the traffic inbound to Core Service is encrypted, consider setting up an IPsec tunnel for use by the various [server-side components of Datazen Enterprise Server](#). Use of an encrypted tunnel will greatly reduce the ability malicious parties to eavesdrop on server-side network traffic.

- Restrict repository backup folder permissions.

When configuring the Core Service [repository backup](#), ensure the target folder's security permissions do not allow unnecessary access. The backup (although encrypted) contains sensitive information that should not be accessible to non-administrative users.

## Web Applications

- Install Web Applications in the DMZ.

Web Applications is the only component of the Datazen Enterprise Server product that features a public interface. Since no sensitive information is ever stored within Web Application state, it may be safely [deployed to a DMZ](#). Other components of the Datazen Enterprise Server should be deployed to the secure internal network.

- Only allow required network traffic from the internal network.

Since Web Applications is the public interface of the Datazen Enterprise Server product it is important to only allow necessary network traffic to reach these machines, such as Remote Desktop traffic (for management purposes).

- Allow only HTTPS traffic from the public network.

Although Web Applications can be [configured to respond to HTTP requests](#), it is strongly recommended that this functionality be relegated in favour of HTTPS.

- Do not modify IIS application pools identities or privileges.

Datazen Enterprise Server Web Applications run under three IIS applications pool identities: *Datazen.Server.ControlPanel*, *Datazen.Server.WebApi* and *Datazen.Server.Renderer*. These application pool identities are created and configured by the [Datazen Enterprise Server installer](#) with reduced security privileges. Granting excessive permissions to these application pool identities could increase the vulnerability to attacks via the Datazen Enterprise Server web interface.

## Data Acquisition Service

- Do not run Data Acquisition Service as LocalSystem.

During the initial installation of [Data Acquisition Service](#) you are prompted for the credentials to run the service as. Choose a local service account, and not LocalSystem. The service account requires the following rights and privileges:

- Modify files in the Datazen Enterprise Server installation target folder.
- Log on as a service.
- Be a member of Performance Log Users group.

The Datazen Enterprise Server installer will configure the required security privileges for the account. LocalSystem has privileges beyond what is required for Data Acquisition Service operation, and can greatly increase risk should this service be compromised.

## Authentication

- Avoid using default authentication.

If possible, avoid using Datazen Enterprise Server default authentication in favour of one of the other supported [authentication mechanisms](#). The default authentication provider stores hashed user credentials within the Core Repository and if compromised, could potentially be used as an attack vector.

If all of the Datazen Enterprise Server users have Active Directory accounts, and the Active Directory environment is accessible to the Datazen Core Service (e.g. the Core Service machine is a domain member, or an Active Directory catalog can be queried via LDAP) consider using Active Directory authentication mode.

If all of the Datazen Enterprise Server users can utilize an existing Active Directory Federation Service environment, consider using ADFS authentication mode.

## Users and Hubs

- Configure BI hubs to adhere to organizational structure.

From a security standpoint, Datazen Enterprise Server [BI hubs](#) should reflect your organizational structure. Although Datazen Enterprise Server users can be placed into multiple BI hubs, members cannot view or interact with dashboards or data residing in BI hubs where they are not members.

- Restrict membership of the hub *Owner* role.

[Owners](#) have special privileges within a [BI hub](#), such as creating and editing data connections and views, and managing membership for the hub and user groups contained within the BI hub. Only allow users who should have these elevated responsibilities to be in the *Owner* role.

## Data Connections

- Only use encrypted connections.

If the data provider allows it, always use an encrypted connection when creating a data connection. Connecting to a back-end data source with an unencrypted connection could expose user credentials to malicious parties.

- Ensure SSAS back-end servers running IIS/msmdpump.dll are secured with SSL.

When configuring [SSAS data connections](#) to back-end servers running IIS/msmdpump.dll, make sure the endpoint is secured with SSL. Connecting to SSAS data sources that are not configured in this way could expose user

credentials to malicious parties.

- Lock down data providers

If the Datazen Enterprise Server data connection requirements are well known, the server administrator should [lock down the set available of data providers](#) so that data connection authors may only select a provider from this approved set.

# Initial Configuration

This section contains the following documents:

- [Authentication](#)
- [Initial Users](#)
- [Hub Configuration](#)
- [Publishing Initial Content](#)

# Authentication

Datazen users can be authenticated by one of the following authentication providers:

- Default
- Active Directory
- ADFS
- External

## Default authentication

The default authentication mode stores and validates user passwords within the Datazen Core Service. When a user is created a notification email is sent to the user which contains a URL to initialize the account and set a password. Users may change their password at any time by managing their profile via the [Control Panel](#).

## Active Directory authentication

Active Directory authentication mode utilizes an existing Active Directory infrastructure for password validation. In this mode, when a user attempts to log in the following validations are performed:

1. The username is checked first within the Datazen Core Service user list.
2. If found the username and password are forwarded to Active Directory for further validation.

In this mode passwords are not stored within the Datazen Core Service Repository.

For more information on configuring Active Directory authentication mode, please refer to the Datazen Enterprise Server [installation documentation](#).

## ADFS authentication

ADFS authentication mode utilizes an existing ADFS server for username and password validation. In this mode, when a user attempts to log in the following actions are performed:

- From a mobile device...
  1. The device queries the Datazen Web Api for the location of the ADFS server.
  2. The device will send the user's credentials to the ADFS server.
  3. The ADFS server will validate the credentials and return a response to the device.
  4. The device will forward the response to the Datazen Web Api.
  5. The Datazen Web Api will check that the user exists in the Datazen Core Service user list.
- From a web browser (either Datazen Web Viewer or Control Panel)...
  1. The browser will be redirected to the ADFS server's login page.
  2. Once authenticated, the browser will be redirected back to the appropriate Datazen site.
  3. The Datazen site will check that the user exists in the Datazen Core Service user list.

In this mode, passwords are not stored within the Datazen Core Service Repository.

**NOTE:** Since the *admin* user can only be authenticated using the default authentication provider, a special url is available allowing *admin* to login to the Control Panel directly, bypassing the ADFS redirection:

`/cp/account/adminlogin`

For more information on configuring ADFS authentication mode, please refer to [Configuring ADFS authentication](#).

## External authentication

External authentication mode is used in scenarios when user credentials are validated *before* a client attempts to connect to Datazen, either through an IIS filter, proxy or some other mechanism. When this mode is active all incoming client HTTP requests are inspected for either a cookie value or header value containing the requester's Datazen username. If this username represents an existing Datazen user the request is allowed to execute.

In this mode, passwords are not stored within the Datazen Core Service Repository.

The external authentication provider can be configured in the [Control Panel Authentication settings](#).

**IMPORTANT:** Since this authentication mode means the Datazen explicitly trusts that external authentication has been performed, it is critically important to ensure that all inbound client traffic to the Datazen public interfaces (*Web Api*, *Control Panel* and *Web Viewer*) be proxied through the external authentication mechanism.

**NOTE:** Since the *admin* user can only be authenticated using the default authentication provider, a special url is available allowing *admin* to login to the Control Panel directly:

`/cp/account/adminlogin`

The external authentication mechanism should account for this special url and provide a means for unauthenticated traffic to reach it.

# Initial Users

## The *admin* user

When Datazen Enterprise Server is initially installed, a special *admin* user is created. This user has full administrative rights to perform any operation in the server instance, and is the only user that can create other users or hubs. When the *admin* user logs in to the Control Panel, options for managing and configuring the server instance will be available.

## The *guest* user

If created, the user with username *guest* has special significance. This account is used for public access to dashboards. When maintaining the *guest* account, ensure that it can only [access](#) dashboards which are intended to be publicly available.

## Server users

Before Datazen BI hubs can be created, at least one initial user must be created. To create the first user, log into the Control Panel as *admin*, and go to *Server Users* to click on *Create User*.

For more information about using the Control Panel to manage user accounts, see [Managing Hub Users and User Groups](#).

# Hub Configuration

Before any content can be published to the server, the first hub must be created and configured.

1. On the *Home* screen of the Control Panel, click on *Create BI Hub*.

Create Hub

Hub name:

Test Hub

Owner username:

Maximum users:

Create

Cancel

Create BI Hub dialog

1. Enter Test Hub for the Hub name.
2. Enter a username previously created for the *Owner username*.
3. Enter 50 for *Maximum users*.
4. Click *Create*. When this is done, click *Done*.

Next, we may want to add more users to the newly created hub.

1. Switch to the new hub using the hub dropdown in the navigation pane on the left-hand side of the Control Panel.

Branding

Log Viewer

Backup

Hub: Test Hub

KPIs

Dashboards

Data Sources

Custom Maps

Test Hub

020

Page 1 of 1

Create User

Batch Create Users

Create D

Hub selector

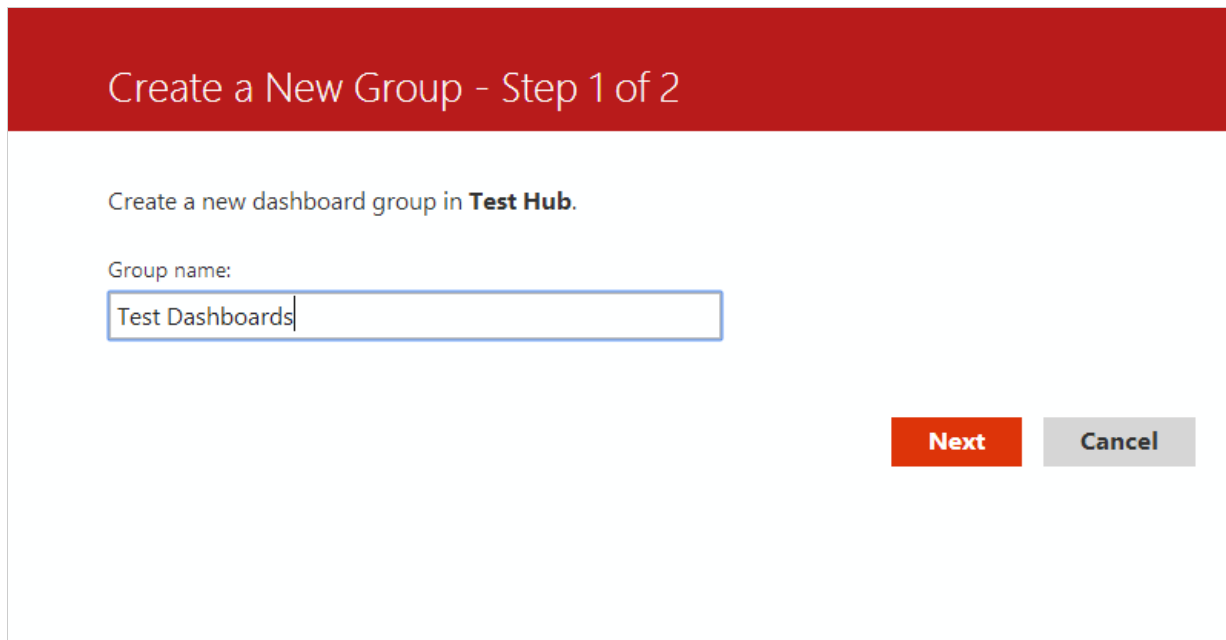


1. Click on *Users*. You should see the hub owner as the sole user in the new hub.
2. Click on *Invite More Users*.
3. Enter the email addresses of users to be added to this hub as a delimited list in the text input area and click *Invite Users*.

Users already registered on the server will automatically be added to the hub. Others will be emailed with instructions on how to join the server and access the hub.

In order to be able to publish dashboards to the new hub, we need to create at least one dashboard group.

1. Click on *Dashboards* on the left-hand side nav.
2. Click on *Create a New Group*.



Create a New Group - Step 1 of 2

Create a new dashboard group in **Test Hub**.

Group name:

Test Dashboards

Next Cancel

Create Dashboard Group dialog

1. Enter Test Dashboards for the name. Click *Next*.

## Create a New Group - Step 2 of 2

Set Permissions:

User Groups



 User Group	Allow Access
 Everyone	<input checked="" type="checkbox"/>

Previous

Create

Cancel

Dashboard Group access permissions dialog

1. Give *Everyone* access rights for now.
2. Click *Create*.

You should now be able to log in to the server from the Publisher application and create and publish a dashboard to the new dashboard group.

## Publishing Initial Content

Once an initial user and hub have been created, we can publish some initial content to the server.

### Create a test KPI

Log in to the Control Panel with the user account previously created. We will first need to create an initial KPI group.

1. Under *KPIs*, click on *Create a New Group*.
2. Select *No Connection* for the data connection. We will manually enter the data for this first KPI. Enter *Test KPIs* for the title. Click *OK*.
3. Give *Everyone* access.

You should now see an empty KPI group named *Test KPIs*. We will create a test KPI in this group.

1. Click on *Create New KPI* on the group header.

The 'New KPI' dialog box is shown. It features a red header with the text 'New KPI'. Below the header, there is a preview section on the left showing a green KPI card for 'Test KPI 1' with a value of '\$123,000 (+11%)' and a bar chart. To the right of the preview are several input fields and dropdown menus. The 'KPI Name' field contains 'Test KPI 1'. The 'Value Format' dropdown is set to 'Currency'. The 'Value' field contains '123000'. The 'Goal' field contains '111000'. The 'Status' dropdown is set to '+1 (green)'. The 'Trend Set' dropdown is set to 'Set manually'. The 'Enter Trend Set value' field contains '40, 25, 5, 23, 45, 50, 55, 70, 61, 56, 63'. There are also dropdown menus for 'Set manually' and 'Enter Value'. At the bottom right, there are 'Create' and 'Cancel' buttons.

New KPI dialog

1. Enter *Test KPI 1* for the *KPI name*.
2. Select *Currency* for the *Value Format*.
3. Enter *123000* for the *Value*.
4. Select *Set manually* for the *Goal*, and enter *111000*.
5. Select *Set manually* for the *Status*, and select *+1 (green)* for its *value*.
6. Select *Set manually* for the *Trend Set*, and accept the default random values.
7. Click *Create*.

You should see the *Test KPI 1* in the group.

## Connect with the Publisher application

Using the Datazen Publisher, connect to the server.

1. On the bottom app menu, click *Connect*.
2. On the *Server Connect* dialog, enter the server address and the credentials for a user you've created.
3. Click *Connect*.

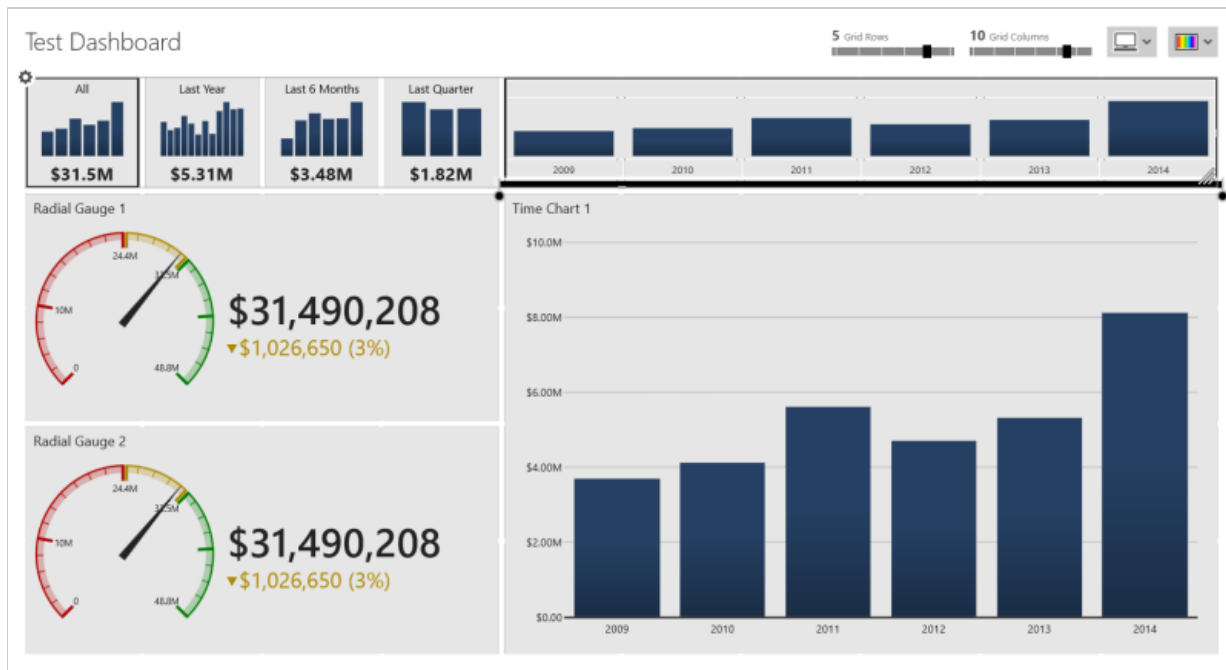
You should now be connected to the server. If you created a test KPI in a previous step, you should now see it on the home screen.

## Publish a test dashboard

To create a simple test dashboard, activate the bottom app menu again and click *Create*.

1. In the dashboard designer, entering the dashboard name at the top.
2. Drag a few dashboard elements onto the design surface: a Time Navigator, two Radial Gauges and a Time Chart.

As you do this, simulated data is generated to produce an approximation of the dashboard's appearance.



Test dashboard

You can click *Run Preview* to see your test dashboard in action. Return to the designer to publish the dashboard to the server.

1. On the bottom app menu, click *Publish to Server*.

First, the dashboard must be saved on the local file system.

Verify the file name and default *local* dashboard group (*My Dashboards* is fine for now) and click *Save*.

The first time you do this, you will be prompted to select a working folder for the publisher. This is the folder where your local dashboards will be stored.

Next, we are prompted for the server destination.

Make sure your server is selected, as well as the right hub and dashboard group. Click *Publish*.

Your test dashboard should now be published on the Datazen server. Back on the home screen, it should appear under *Test Dashboards*.

## **Verify initial content using Web Viewer**

The content you have created is also accessible via the Web Viewer, at *http(s)://your.server.address/viewer*. Use your browser to navigate there.

Log in using the credentials you used to publish your first dashboard. You should see your initial content.

Clicking on the dashboard tile should present the interactive web view of the dashboard.

Use the top menu to switch between dashboards and KPIs.

For more information on the Web Viewer, see the [Web Viewer documentation](#).

# Using the Control Panel

This section contains the following documents:

- [Managing KPIs](#)
- [Managing Dashboards](#)
- [Data Access](#)
- [Data Source Types](#)
- [Managing Custom Maps](#)
- [Managing Hub Users & User Groups](#)
- [Managing Permissions](#)
- [Managing Custom Branding](#)
- [Dashboard Runtime](#)

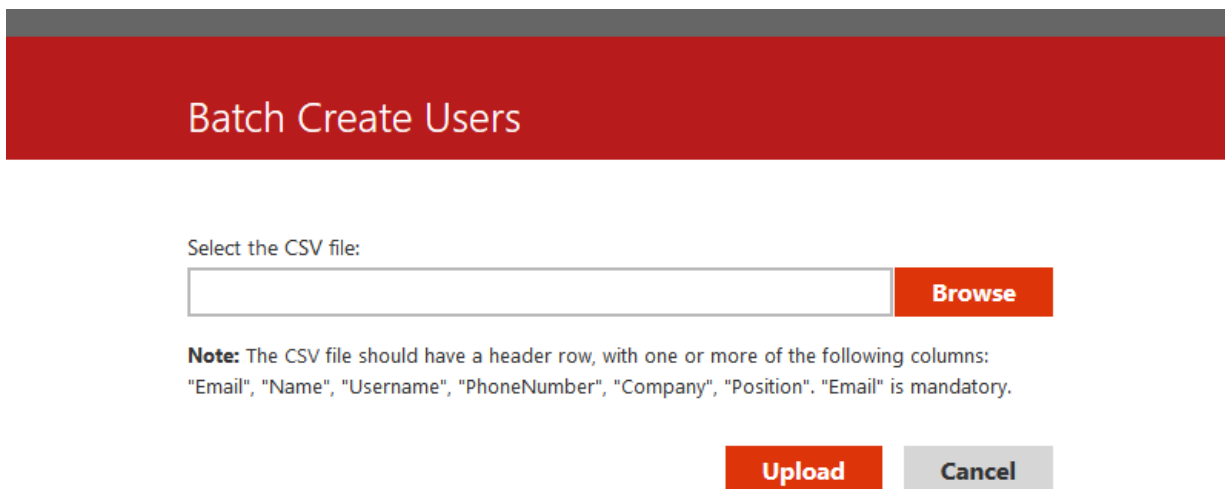
## Batch Creating Users

To create more than one user at a time, a comma-separated value (CSV) file containing the user details for a set of new users can be uploaded. The CSV file must contain a header row and one or more data rows. The header row/column values are:

- Email (mandatory)
- Username (mandatory)
- Name (mandatory)
- PhoneNumber (optional)
- Company (optional)
- Position (optional)

For example:

```
Email,Username,Name
michael.z.madsen@gmail.com,mmadsen,Michael Z. Madsen
linda.b.lacelove,llacelove,Linda B. Lacelove
```



Batch Create Users

Select the CSV file:

**Browse**

**Note:** The CSV file should have a header row, with one or more of the following columns: "Email", "Name", "Username", "PhoneNumber", "Company", "Position". "Email" is mandatory.

**Upload** **Cancel**

The Batch Create Users dialog.

1. Navigate to *Home*. Click on *Batch Create Users*.
2. Click on *Browse* in the dialog that appears; select the file to be uploaded.
3. Click *Upload*.
4. Click *OK*.

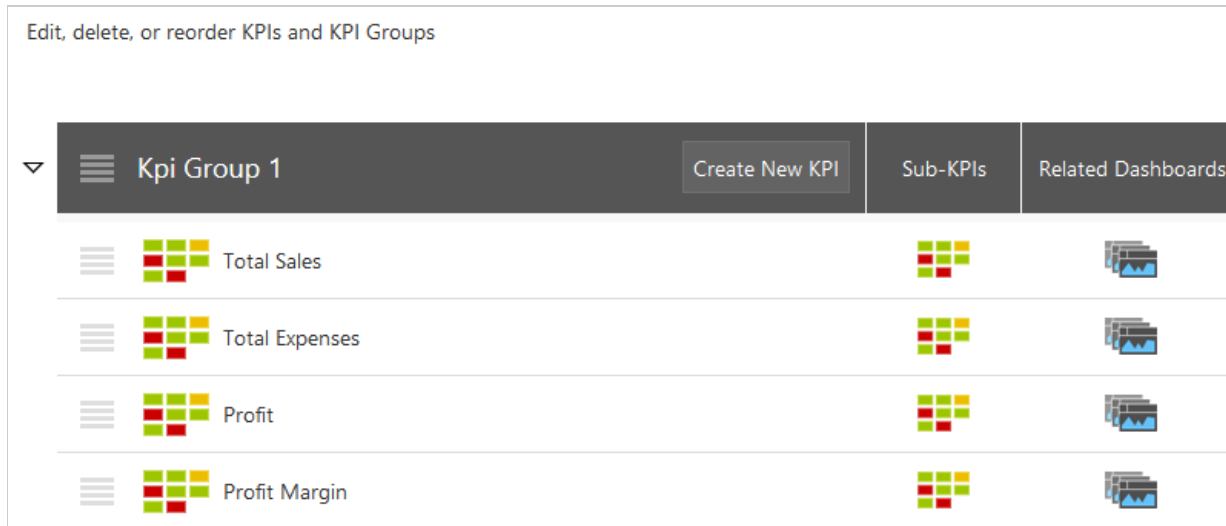
The users have now been added to the system.

# Managing KPIs

The Control Panel *KPIs* page allows users to create and organize KPIs in a hub.

## KPI groups

KPI groups contain information on how KPIs will connect to remote data. Unlike [Data Connections](#), they can be backed by static values instead of remote data.



KPI edit dialog.

## Creating a KPI group

1. Click on *Create a New Group*.
2. Select the desired KPI group type and complete the form. See [Data Source Types](#) for more information.
3. Fill out the name and other properties for the connection. Click *Test Connection* to verify that the data provider can be reached by the Datazen server. Click *Next* when done.
4. Grant access permissions to user groups or users. Access permissions can be set at any time from the [Permissions page](#).
5. Click *Finish* when done.

The new KPI group will now be listed among the available KPI groups.

## Editing a KPI group

1. Click the *Edit* icon for the KPI group.
2. Update the properties in the presented dialog.
3. **Optional** Click *Test Connection* to verify the connection status.
4. Click *Next* when done.
5. Update the permissions. Click *Finish* when done.

The KPI group will now be updated.

## Removing a KPI group

1. Click the *Delete* icon for the KPI group.
2. Click *Delete* in the presented dialog.



The KPI group will now be removed from the system.

## KPIs

KPIs are defined and managed in the Control Panel. Each KPI can consist of a primary setting: Value, as well as optional secondary settings:

- Goal, which is used for comparison against the Value
- Status, which defines the background of the KPI visual
- Trend Set, which defines the graph of the KPI

Each of these settings can be set manually (as a static value), or be defined in a query which will be applied against the KPI group's data connection. These queries, like [Data Views](#), may be [personalized](#).


KPIs can have child sub-KPIs, which themselves are full KPIs, inheriting the KPI group data connection. KPIs can also be related to one or more dashboards, which will be presented when the KPI is viewed in a client application.

New KPI

Preview:

Test KPI 1

\$123,000 (+11%)



KPI Name:

Test KPI 1

Value Format:

Currency

Value:

Set manually

Enter Value:

123000

Goal:

Set manually

Enter Goal value:

111000

Status:

Set manually

Enter Trend value:

+1 (green)

Trend Set:


Set manually


Enter Trend Set value:


40, 25, 5, 23, 45, 50, 55, 70, 61, 56, 63


Visualization:

NONE









Create

Cancel

KPI and KPI groups listing.

### Creating a KPI

1. Scroll to the *KPI group* to which the new KPI should be added. If this will be a sub-KPI, navigate down the *Sub-KPIs* link from the parent KPI.
2. Click *Create New KPI* on the KPI group/KPI. A form presenting the KPI properties will appear.
3. Complete the form. The *KPI Name* must be unique within the parent KPI group.
4. Click *Create* to complete the KPI creation process.

The new KPI will now be listed among the other KPIs.

## Changing a KPI's related dashboards

1. Navigate to the KPI via the sub-KPIs if necessary.
2. Click the *Related Dashboards* icon for the KPI.
3. Add related dashboards by clicking them in the left column; reorder them by dragging the dashboard in the right column; remove them by clicking the *X*.
4. Click *Update* when done.

The KPI's related dashboards will now be updated.

## Edit a KPI

1. Navigate to the KPI via the sub-KPIs if necessary.
2. Click the *Edit* icon for the KPI.
3. Update the properties in the presented dialog. Click *Update* when done.

The KPI will now be updated.

## Remove a KPI

1. Navigate to the KPI via the sub-KPIs if necessary.
2. Click the *Delete* icon for the KPI.
3. Click *Delete* in the presented dialog.

The KPI has now been removed from the system.

# Managing Dashboards

The Control Panel *Dashboards* page enables hub owners to reorder the hub’s existing dashboards and dashboard groups, create new dashboard groups, edit the names of dashboards or dashboard groups, delete dashboards or dashboard groups, or control a dashboard’s visibility in a client application’s main screen.

Edit, delete, or reorder dashboards and dashboard groups		
▼	Public Dashboards	Hide on Main Screen
	Sales vs. Expenses	<input type="checkbox"/>
	Sales by Region	<input type="checkbox"/>
	Sales by Industry	<input type="checkbox"/>
	Product Scores	<input type="checkbox"/>

Dashboards and dashboard groups listing.

## Dashboards

### Renaming a dashboard

1. Click on the *Edit* icon next to the dashboard.
2. Update the name of the dashboard in the dialog that appears.
3. Click *Apply* when done.

The dashboard has now been renamed.

### Deleting a dashboard

1. Click on the *Delete* icon next to the dashboard.
2. Click *Delete* when prompted.

The dashboard has now been deleted.

### Reorder/move a dashboard

1. Click on the drag-handle of the dashboard, and drag the dashboard to its new position. The dashboard can also be dragged into a different dashboard group.
2. Release the drag-handle.

The dashboard has now been reordered/moved.

### Hide a dashboard

1. Click on *Hide on Main Screen* checkbox for the dashboard.

The dashboard is now hidden from the main screen of client applications. This is intended to prevent users from directly viewing dashboards that are intended to only be accessed via [drill-through](#).

## Dashboard groups

### Creating a dashboard group

1. Click the *Create a New Group* button.
2. Enter a group name in the dialog that appears. Click *Next* to continue.
3. Grant access permissions to user groups or users. Note that these access permissions can be set at any time from the [Permissions page](#).
4. Click *Create* to complete the process.

The dashboard group will now be added to the selected hub.

### Renaming a dashboard group

1. Click on the *Edit* icon next to the dashboard group.
2. Update the name of the dashboard group in the dialog that appears.
3. Click *Apply* when done.

The dashboard group has now been renamed.

### Deleting a dashboard group

1. Click on the *Delete* icon next to the dashboard group.
2. Click *Delete* when prompted.

The dashboard group, and any dashboards it may have contained, has now been deleted.

### Reorder a dashboard group

1. Click on the drag-handle of the dashboard group, and drag the group to its new position.
2. Release the drag-handle.

The dashboard group has now been reordered.













# Data Access Concepts

Datazen dashboards use tabular data (*data views*) to power their visualizations. Datazen Enterprise Server data are configured on the server and backed by various data providers (*data connections*). This data can then be added to visualizations by the dashboard author, having connected to the Datazen Server. Without a connection to Datazen Enterprise Server, data is limited to local Microsoft Excel spreadsheets that are available to the author when the dashboard is created.

## Data connections

All Folders and Data Connections

New Folder...

Folder/Data Connection	Edit	Delete
 <b>Marketing Documents</b>		
 <b>Operations Documents</b>		
 <b>Corporate Intranet SharePoint lists</b>		
 <b>Sql Azure DemoDb</b>		



















Data connection listing.

Data connections contain information (i.e. address, credentials) about how to connect to remote data so that it can be retrieved and be made available to dashboard authors. Depending on the [data provider](#), different configuration settings are available.

Only hub owners are allowed to create or manage data connections. [Access permissions](#) must be set on a data connection before other hub members are allowed to use it.

Many different [data providers](#) are available for use with data connections. The set of data providers can be extended by [creating custom data providers](#). Also, built-in data providers can be [administratively disabled](#) so they will not be available to data connection authors.

## Data views

  <span>Sql DemoDb</span>					
Data View		Edit	Delete	View	Status
	Daily Sales				Last Update: 11:11 Next Update: 12:00 Last Status: <span>Suc</span>
	Expense Category Lookup				Last Update: 11:11 Next Update: 12:00 Last Status: <span>Suc</span>
	Expense Time Lookup				Last Update: 11:11 Next Update: 12:00 Last Status: <span>Suc</span>
	Expenses By Category LOD				Last Update: 7:11 Next Update: 7:00 Last Status: <span>Suc</span>

Data view listing.

Some [data connections](#) support data views. A data view uses the parent data connection to connect to the remote data, but has its own settings for selecting which data to acquire from the connection. For example: A Microsoft SQL Server data connection would contain a connection string to locate the data store, and its data view would contain a SQL statement that returns a specific table of data.

Data connections that support data views must have one or more data views defined in order to begin acquiring remote data.

Data views inherit the access permission of the parent data connection. A hub member with access to the data connection will have access to all of the data views contained within it.

**Important:** It is possible to author a dashboard that a hub member has access to, but contains data that the hub member does not have access to. In this scenario the dashboard would display in a Datanen client application but without any data.


## Cached data views

Edit Data View - Step 1 of 2


Data View Name:

Refresh Frequency:

Every 60 minutes
▼

[Define parameters...](#)


☐ Allow Client Data Caching

☐ Personalize for each member


Data View Query:

```
SELECT [Date] ,[Amount] FROM [dbo].[SalesByDay]
```

Editing a data view.

Once data connections and views have been configured in a hub, the Datazen Enterprise Server will [periodically acquire the data from the original data source and store it internally](#), caching it for client use. The schedule for how often the data is acquired is configurable either for the individual data view (for [data connections](#) that support data views) or on the data connection itself.

Data for views can be manually refreshed at any time via the Control Panel's *Data Sources* page.

## Real time data views

The data acquisition schedule can be bypassed by setting the real time flag on the data connection or view. By turning on [real time data retrieval](#), every time a dashboard that utilizes the data is viewed the original data store is re-queried for the data.

Real time views may use [parameters](#) in their query to filter data before it is sent to a dashboard.

**Important:** Hub owners should be aware of the increased network utilization incurred for real time data, particularly where the original data resides over a metered connection or in a fee-based service such as Windows Azure.

## Personalized data views






By default, the data provided to a specific dashboard by a data connection or view is the same for all users. [Data personalization](#) allows this data to be specific to each user. When a dashboard is viewed that utilizes personalized data, the server will return only the data specific to the logged-in user.

When this feature is enabled the data retrieval mechanism will issue and store a personalized data item for every user that has been granted [explicit access permission](#) to the data connection. Note that data is not retrieved when the access permission are modified, but according to the defined data refresh interval.

The data personalization feature usage is different for the various data connections that support it. For more information, please see the [data connection reference](#).

## Parameterized data views

Define Parameters

Name	Type	Default Value	Edit
@categoryCode	Text	RU	
@startTime	Date / Time	2009-01-01	
@endTime	Date / Time	2010-01-01	
<input type="text" value="@"/>	<input type="text" value="Text"/> 	<input type="text" value="Default value"/>	

Defining data view parameters.

Real time views may use [parameters](#) to filter data before it is sent to a dashboard, which would otherwise retrieve the full set of data. These parameters can be of type *Text*, *DateTime* or *Number*. A default value can also be defined for

cases where the parameter is not set or unavailable.

When a dashboard is built using a real time view with parameters, the designer will provide the user with the option to bind the dashboard's filterable properties to these parameters.



# Data Providers

Hub owners can make data available to dashboard or KPI authors by creating server data connections from one of the available data providers. For more information on extending data providers, see [Managing Data Providers](#).

## Built-in data providers

### Microsoft SQL Server

Accesses data stored in a Microsoft SQL Database.

Properties:

- *Connection Properties* - A panel of connection properties are used to construct a connection string to the Microsoft SQL Server database.

View properties:

- *Data View Query* - The SQL query to perform. Must return a single table of data.

### Microsoft SQL Server Analysis Services

Accesses data stored in a Server Analysis Services Server.

Properties:

- *Connection Properties* - A panel of connection properties are used to construct a connection string to the Microsoft SQL Server Analysis Service. The connection string can access the SSAS server directly, or [via HTTP](#). If connection via HTTP, please ensure the SASS instance has been configured for [IIS/MSMDPUMP](#).
- *Personalize with Effective User* - Whether to personalize data views at the connection level using per-user authentication.

View properties:

- *Data View Query* - The MDX query to perform. Must return a single table of data. Any [Dimension hierarchy](#) which needs to be returned as part of the view must be expressed as a defined [Member](#) in the MDX Query.

### Microsoft SharePoint List

Accesses data stored in a Microsoft SharePoint list.

Properties:

- *Server* - The base URL of the [SharePoint 2010 REST service](#) or [SharePoint 2013 REST service](#).
- *Username* - The username of the SharePoint user accessing the list.
- *Password* - The password of the SharePoint user accessing the list.

View properties:

- *List Name* - The name of the SharePoint list to retrieve.

## Microsoft Azure SQL Database

Accesses data stored in a Microsoft Azure SQL Database.

Properties:

- *Connection Properties* - A panel of connection properties are used to construct a connection string to the Microsoft Azure SQL Database instance.

View properties:

- *Data View Query* - The SQL query to perform. Must return a single table of data.

## OData (Open Data Protocol) Web Service

Accesses data returned from an online [OData](#) web service.

Properties:

- *Basic Authentication credentials* (optional)

View properties:

- *OData view URL* - The URL of the OData web service.

## Generic XML Web Service

Performs an HTTP request for a single XML document of data.

Properties:

- *URL of XML Web Service* - The URL of the web service return the XML data. Generally, the XML data should be in a form that can be loaded into a .NET DataSet.
- *Basic Authentication credentials* (optional)

## Generic ODBC DSN

Accesses data returned though a system ODBC DSN.

**IMPORTANT:** The DSN must be registered on the Datazen Core Service machine and Data Acquisition machine(s).

Properties:

- *Connection Properties* - A panel of connection properties are used to construct an ODBC connection string to system ODBC DSN.

View properties:

- *Data View Query* - The query to perform. Must return a single table of data.

## MySQL ODBC Driver

Accesses data stored in a MySQL Database via an official ODBC driver.

**IMPORTANT:** The selected driver must be installed on the Datazen Core Service machine and Data Acquisition machine(s).

Properties:

- *Connection Properties* - A panel of connection properties are used to construct an ODBC connection string to the MySQL Database.

View properties:

- *Data View Query* - The SQL query to perform. Must return a single table of data.

## Oracle ODBC Driver

Accesses data stored in an Oracle Database via the official ODBC driver.

**IMPORTANT:** The driver must be installed on the Datazen Core Service machine and Data Acquisition machine(s).

Properties:

- *Connection Properties* - A panel of connection properties are used to construct an ODBC connection string to the Oracle Database.

View properties:

- *Data View Query* - The SQL query to perform. Must return a single table of data.

## PostgreSQL ODBC Driver

Accesses data stored in a PostgreSQL Database via an official ODBC driver.

**IMPORTANT:** The selected driver must be installed on the Datazen Core Service machine and Data Acquisition machine(s).

Properties:

- *Connection Properties* - A panel of connection properties are used to construct an ODBC connection string to the PostgreSQL Database.

View properties:

- *Data View Query* - The SQL query to perform. Must return a single table of data.

## Oracle Provider for OLE DB

Accesses data stored in an Oracle Database via the *OraOLEDB.Oracle* OLE DB driver.

**IMPORTANT:** The driver must be installed on the Datazen Core Service machine and Data Acquisition machine(s).

Properties:

- *Connection Properties* - A panel of connection properties are used to construct an OLE DB connection string to the Oracle Database.

View properties:

- *Data View Query* - The SQL query to perform. Must return a single table of data.

## PGNP OLEDB Provider for PostgreSQL

Accesses data stored in a PostgreSQL Database via the *PGNP* OLE DB driver.

**IMPORTANT:** The driver must be installed on the Datazen Core Service machine and Data Acquisition machine(s).

Properties:

- *Connection Properties* - A panel of connection properties are used to construct an OLE DB connection string to the PostgreSQL Database.

View properties:

- *Data View Query* - The SQL query to perform. Must return a single table of data.

## Excel Documents in a Network Share

Reads Microsoft Excel documents from a local folder or network share.

Properties:

- *Path on Server* - The fully qualified local or remote (UNC) path *from the server* to the folder containing Microsoft Excel documents. The folder or share must be accessible by the Datazen Enterprise Server account.

## Supported Features

This matrix describes additional features supported by the different data source types:

Data Providers	Has Data Views	Supports Parameterization	Supports Personalization	Supports Real-time
MS SQL	Yes	Yes	Yes	Yes
MS SSAS	Yes	Yes	Yes	Yes
MS SharePoint List	Yes	Yes	Yes	Yes
MS SQL Azure	Yes	Yes	Yes	Yes
OData Service	Yes	Yes	Yes	Yes
XML Web Service		Yes	Yes	Yes
ODBC DSN	Yes	Yes	Yes	Yes
ODBC MySQL	Yes	Yes	Yes	Yes








ODBC Oracle	Yes	Yes	Yes	Yes
ODBC PostgreSQL	Yes	Yes	Yes	Yes
OLE DB Oracle	Yes	Yes	Yes	Yes
OLE DB PostgreSQL	Yes	Yes	Yes	Yes
Excel Network Share				Yes

# Managing Custom Maps

Map files (as used by the Datazen map controls) can be uploaded and stored on a hub via the Control Panel's *Custom Maps* page.

All Folders and Map Shapes

New Folder...

Folder/Map	Edit	Delete
 Custom		
 california		
 ukraine		

Custom map listing.

## Upload a map

1. Click on the "Upload Map Shape..." button.
2. Type a name for the map.
3. Select a shape (.SHP) and attribute (.dbf) file.
4. Click *Upload*; the dialog will close once the upload is complete.

The custom map is now available to users connected to the hub.

## Removing a map

1. Navigate to the map via the folders if necessary.
2. Click the *Delete* icon for the map.
3. Click *Delete* in the presented dialog.

The map will now be removed from the system.

## Moving a map

1. Navigate to the map via the folder if necessary.
2. Click the *Move* icon for the map.
3. Navigate to the destination folder in the presented dialog. Click *Move* when done.

The map will now appear in the new location.

# Managing Hub Users and User Groups

A hub owner (or server admin) may configure hub membership in the Control Panel's *Users* page. User groups provide a way to apply permissions to sets of users, and can be administrated in the *User Groups* page.

Datazen Enterprise Server users are created via the Control Panel. At least one user **must** be added to the server before the initial hub can be created. Users must be added to hubs before they can access or publish dashboards or other hub content.

## The *admin* user

When Datazen Enterprise Server is first installed, a special *admin* user is created. This user has full administrative rights to perform any operation in the server instance, and is the only user that can create other users or hubs.

When logged in to the Control Panel as the *admin* user, several options for managing and configuring the server instance are available.

## The *guest* user

If created, the user with username *guest* has special significance. This account is used for public web access to dashboards and KPIs. When maintaining the guest account, ensure that it can only access those dashboards and KPIs which are intended to be viewed publicly.

Search Users:

#	Username	Email	Is Publisher	Is Owner
1	<b>breon</b>	breon@componentart.com	<input type="checkbox"/>	<input type="checkbox"/>
2	<b>brian</b>	brian@componentart.com	<input type="checkbox"/>	<input type="checkbox"/>
3	<b>chris</b>	chris@componentart.com	<input type="checkbox"/>	<input type="checkbox"/>
4	<b>corey</b>	corey@componentart.com	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Users listing.

## Adding users to a hub

1. Navigate to the *Users* screen. Click *Add Users*.
2. Enter usernames (delimited by space, comma, semicolon, or new line) of the users to be added to the current hub.
3. Click *Add Users to [current hub]*.

An e-mail will be sent to the user notifying them that they have been added to the hub.

## Removing users from a hub

1. Navigate to the *Users* screen.

2. Find the user(s) to be removed, either through paging the table or by searching for them via the *Search Users* search box.
3. Click the *Remove User* icon next of the user(s).
4. Click *Remove* when prompted.

The user(s) will now be removed from the hub.

## Assigning publishers









1. Navigate to the *Users* screen.
2. Find the user(s) to become a publisher, either through paging the table or by searching for them via the *Search Users* search box.
3. Click on the *Is Publisher* checkbox for that user(s).

The user(s) will now be a publisher for the hub.

## Assigning hub ownership

1. Navigate to the *Users* screen.
2. Find the user to become the new owner, either through paging the table or by searching for them via the *Search Users* search box.
3. Click on the *Is Owner* checkbox for that user.
4. Click *Add* when prompted.

The user is now a hub owner; hubs always have at least one owner.

Edit, delete, or create user groups:		
 User Group	Edit Group Name	Delete Group
 Everyone		
 Marketing	 Abc	
 Sales	 Abc	

User groups listing.

## Creating a user group

1. Navigate to the *User Groups* screen. Click *Create New User Group*.
2. Enter a name for the group when prompted. The name should be unique among user groups in the hub.
3. Click *Create*.

The user group will be added to the hub, and should appear in the list.

## Managing user group membership



1. Navigate to the *User Groups* screen.
2. Click *Users...* for the user group to be managed.
3. To add users to the user group, click their checkbox(es) in the table on the left and then click *Add*. To remove users from the user group, click their checkbox(es) in the table on the right and click *Remove*.

The users will be added/removed when the button is clicked, without prompting.

## Removing a user group

1. Navigate to the *User Groups* screen.
2. Click the *Delete Group* icon for the user group.
3. Click *Delete* when prompted.

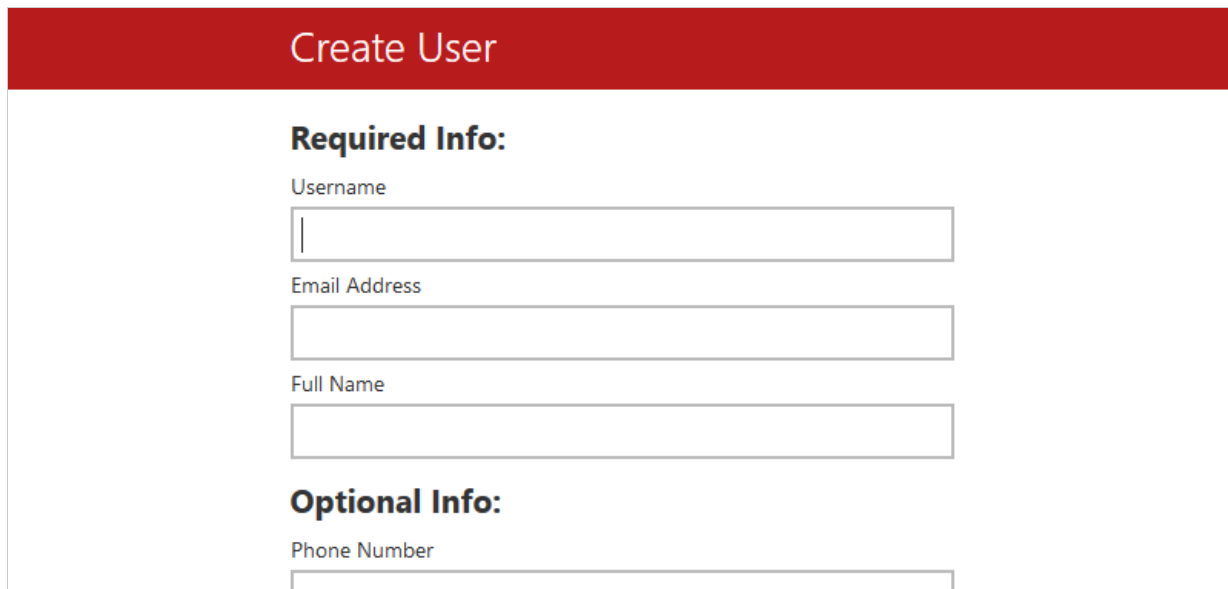
The user group will now be removed from the hub.

## Renaming a user group

1. Navigate to the *User Groups* screen.
2. Click the *Edit Group* icon for the user group.
3. Update the name of the user group in the appropriate text box.
4. Click *Apply* when done.

The user group has now been renamed.

The Datazen *admin* user may create or remove users within the Control Panel. At least one user must be created before the initial hub may be created.

The image shows a 'Create User' dialog box. It has a red header bar with the text 'Create User' in white. Below the header, the form is divided into two sections: 'Required Info:' and 'Optional Info:'. Under 'Required Info:', there are three text input fields labeled 'Username', 'Email Address', and 'Full Name'. Under 'Optional Info:', there is one text input field labeled 'Phone Number'. The form is set against a light gray background with a white border.

Create User dialog.

## Creating a user

1. Click the *Create User* button on the home page.
2. Fill out the user details form. *Username*, *Email* and *Full Name* are required fields. The *Username* and *Email* must be unique within the server instance.  
Note: If the server is operating in Active Directory authentication mode, the username must match the user's Active Directory username.

3. Click *Create User*.

The user has now been added to the system.

- If the server is using [default authentication](#), the user will receive an e-mail notification (see below). They must click the supplied link in order to complete the account creation process.
- If the server is using [Active Directory authentication](#), the user can use the system as soon as they become a member of a hub.

## Batch creating users

See [Batch Creating Users](#).

## Deleting a user

1. Navigate to the *Server Users* screen.
2. Find the user to be deleted, either through paging the table or by searching for them via the *Search Users* search box. Click the *Delete User* icon next of the user.
3. Click *Delete* when prompted.

The user is now deleted from the system.

## Locking/unlocking a user's account

1. Navigate to the *Server Users* screen.
2. Find the user to be locked/unlocked, either through paging the table or by searching for them via the *Search Users* search box. Click the *Account locked* checkbox.
3. Click *Update*.

The user's account is now locked/unlocked. Locked accounts will be denied access to the system.

## Managing the user profile

1. Log in as the user. Click the *Account* dropdown in the upper-right-hand corner.
2. Click *Edit* from the presented options.
3. Update text boxes as necessary.
4. Click *Update My Account*.

The user's profile is now updated.

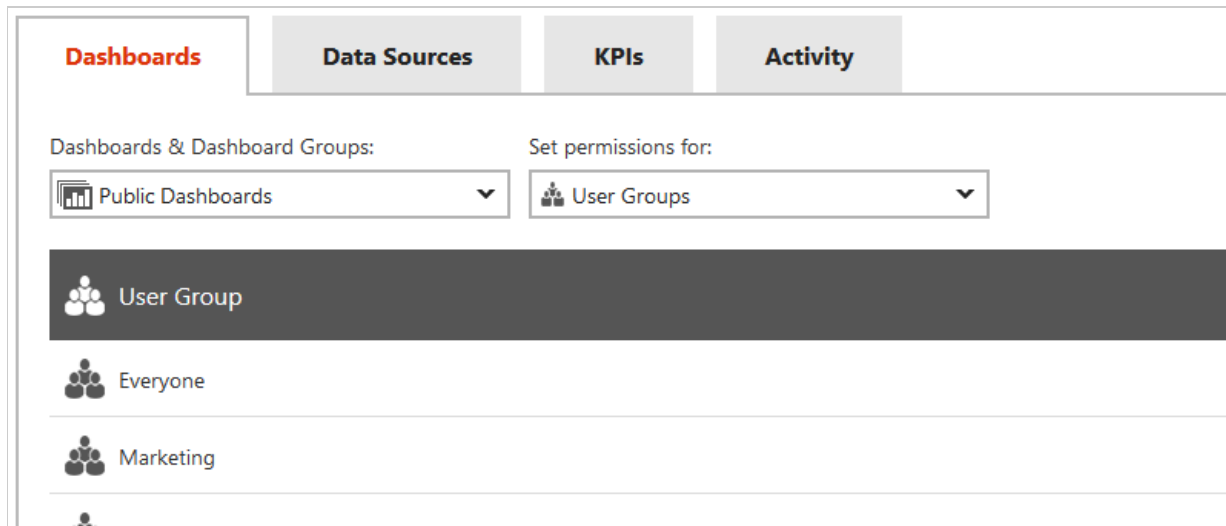
## Email notifications

Control Panel uses SMTP e-mail to notify users when they've been invited to the system, have been added to a hub, or require a new password. These e-mails use templates which can be edited in [Email Templates](#). The SMTP server settings can be updated in [Control Panel's configuration](#).

# Managing Permissions

The Control Panel *Permissions* page enables the hub owner to manage user and/or user group access permissions on:

- Dashboards
- Dashboard groups
- Data connections
- KPI groups
- Activity



Permissions page.

## Change permissions on a dashboard or dashboard group

1. Select the dashboard or dashboard group in the *Dashboards & Dashboard Groups* dropdown.
2. Update the permissions by toggling the *Allow Access* checkboxes. Use the second dropdown to switch between user groups and users.

Changes are persisted immediately.

## Change permissions on a data connection

1. Click the *Data Sources* tab.
2. Select the data connection in the *Folders & Data Connections* dropdown.
3. Update the permissions by toggling the *Allow Access* checkboxes. Use the second dropdown to switch between user groups and users.

Changes are persisted immediately.

## Change permissions on a KPI Group

1. Click the *KPIs* tab.
2. Select the KPI group in the *KPI Groups* dropdown.
3. Update the permissions by toggling the *\*Allow Access* checkboxes. Use the second dropdown to switch between user groups and users.

Changes are persisted immediately.

## Change permissions on activity













1. Click the *Activity* tab.
2. Update the permissions by changing the *Access Level* dropdown.
3. *No Access* hides activity from the user/user group.
4. *Read Only* allows the user/user group to see activity.
5. *Full Access* allows the user/user group to add new comments, or update or remove their old comments.

Use the second dropdown to switch between user groups and users.

Changes are persisted immediately.

# Managing Custom Branding

Branding refers to the graphical resources used by the Datazen client applications when drawing their user interface, such as the background image and color of buttons. The Control Panel *Branding* page allows the *admin* user to assign a default brand for each hub, as well as a default brand for the server instance.

Server/Hub	Branding	Upload	
 Server			
 Corporate Overview			
 Examples			
 KPI Testing			
<div>  Page 1 of 1  </div>			

Branding listing.

## Upload a branding

1. If uploading a brand for a hub, page to the hub using the paging controls.
2. Click on the *Upload* icon for the hub (or Server).
3. Click *Browse* to find and select the branding package on the local file system.
4. Click *Upload*; the dialog will close once the upload is complete.

The branding package will be applied to dashboards and clients connected to this hub/server.

## Removing a branding

1. If deleting a brand for a hub, page to the hub using the paging controls.
2. Click the *Remove* icon for the hub (or Server).
3. Click *Delete* in the presented dialog.

The branding will now be removed from the system.

## Download a branding

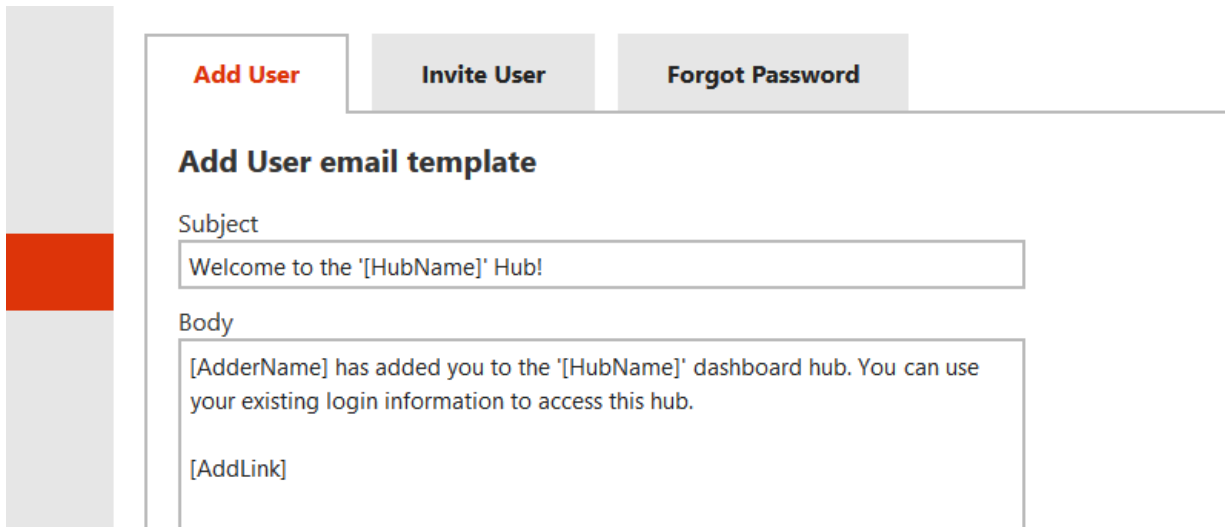
1. If downloading a brand for a hub, page to the hub using the paging controls.
2. Click the *Download* icon for the hub (or Server).
3. Accept the file from the browser's download dialog.

The branding package will now be downloaded to the local file system.

# Managing E-mail Templates

Control Panel's *Email Templates* page is used to edit the subject and body of e-mails that are sent to users and prospective users.

- The “Add User” template is used for e-mails sent to existing users (i.e. have already registered with the system), notifying them that they have been added to a Hub.
- The “Invite User” template is used for e-mails sent to new users (i.e. have not yet registered), to begin their registration process.
- The “Forgot Password” template is used for e-mails sent to existing users who have forgotten their password and need a new one.



The screenshot shows the 'Email Templates' page with three tabs: 'Add User' (selected), 'Invite User', and 'Forgot Password'. Below the tabs, the 'Add User email template' section is visible. It contains two text input fields: 'Subject' with the value 'Welcome to the '[HubName]' Hub!' and 'Body' with the value '[AdderName] has added you to the '[HubName]' dashboard hub. You can use your existing login information to access this hub.' followed by '[AddLink]'.

The Email Templates page.

## Update the Add User template

1. Navigate to *Email Templates*.
2. Update the subject and body text. You can define where the user's name, the hub name, and link appear using this syntax:
3. [AdderName]: The name of the member who added the recipient.
4. [AddLink]: The URL that the recipient should visit in order to continue the adding process.
5. [HubName]: The name of the hub.
6. Click “Update Templates” when done. To cancel your changes navigate to another section of the Control Panel site.

The Add User template has now been updated, and subsequent e-mails sent to added users will use this version.

## Update the Invite User template

1. Navigate to *Email Templates*.
2. Click the *Invite User* tab.
3. Update the subject and body text. You can define where the user's name, the hub name, and link appear using this syntax:
4. [InviterName]: The name of the member who invited the recipient.
5. [InviteLink]: The URL that the recipient can visit to see the hub.
6. [HubName]: The name of the hub.

7. Click "Update Templates" when done. To cancel your changes navigate to another section of the Control Panel site.

The Invite User template has now been updated, and subsequent e-mails sent to invited users will use this version.

## **Update the Forgot Password template**

1. Navigate to *Email Templates*.
2. Click the *Forgot Password* tab.
3. Update the subject and body text. You can define where the link appears using this syntax:
4. [ForgotPasswordLink]: The URL that the recipient should visit in order to continue the password reset process.
5. Click "Update Templates" when done. To cancel your changes navigate to another section of the Control Panel site.

The Forgot Password template has now been updated, and subsequent e-mails sent to users will use this version.

# Managing the Dashboard Runtime

The Dashboard Runtime is a set of files utilized by the Web Viewer and the mobile clients to render dashboards in HTML5. The runtime is installed with Datazen and can be updated at any time with the Control Panel.

## Updating the Dashboard Runtime

Updating the Dashboard Runtime is performed in the Control Panel *Configuration* section. The *Dashboard Runtime* tab will display the currently installed version and provide an option to upload a different version:

<b>Datazen Server</b>	<b>Dashboard Runtime</b>	<b>Data Connections</b>
Current version: <b>3.0.2801</b>		
<b>Upload</b>		

Clicking the *Upload* button will display the upload dialog. Click *Browse* to and pick a new version of the Dashboard Runtime zip file, then click *Upload*:

Upload Dashboard Runtime

Select the dashboard runtime (.zip) file:

**Browse**

**Upload**

**Cancel**



## Using the Web Viewer

The Web Viewer provides access to Datazen dashboards and KPIs via a web browser. The following minimum web browser versions are supported:

Web Browser	Version
Internet Explorer	9
FireFox	3.5
Chrome	3
Safari	4
Opera	10

On mobile devices running iOS, Android or Windows Phone, native apps are the optimal way to access Datazen content.

Note: Cookies and JavaScript must be enabled.

- [Accessing Dashboards and KPIs](#)
- [Configuring Public Access](#)
- [Configuring Integrated Windows Authentication](#)
- [Embedding Datazen Content Into Custom Apps](#)
- [Embedding Datazen Content Into SharePoint](#)

# Accessing Dashboards and KPIs

The Web Viewer is available by browsing the `/viewer` folder of Datazen Enterprise Server's base URL (e.g. `http://local.server.com/viewer`). It is a secure portal and requires users to log in with valid Datazen Enterprise Server credentials. All dashboards and KPIs visible to the logged-in user are then presented for viewing.

The top navigation bar can be hidden by adding the parameter `?header=false` to the query string. It is also possible to link directly to the dashboards or KPIs lists, at `/viewer/dashboards` and `/viewer/kpis`, respectively. For further filtering example, see below. Note that all URLs are case-insensitive.

## Examples of tailored views

To link to...	Use url pattern
public KPIs while omitting the navigation header	<code>http://local.server.com/viewer/public/kpis?header=false</code>
dashboards in the hub <i>Marketing</i> and the group <i>Regional Overviews</i>	<code>http://local.server.com/viewer/dashboards?hub=Marketing&amp;group=Regional%20Overviews</code>
a single specific KPI tile without navigation header	<code>http://local.server.com/viewer/kpis?hub=Sales&amp;group=Salespeople&amp;name=Julie%20Tam</code>
directly to a named dashboard in a specific group	<code>http://local.server.com/viewer/dashboard?hub=Marketing&amp;group=Regional%20Overviews&amp;name=Middle%20East</code>
dashboards accessible to the logged-in Windows domain user	<code>http://local.server.com/viewer/user/dashboards</code>

## Full list of supported URL patterns

- `/viewer`
- `/viewer/Login`
- `/viewer/Home`
- `/viewer/Dashboards`
- `/viewer/Dashboards?hub=Hub&group=Group`
- `/viewer/Dashboard?hub=Hub&group=Group&name=Name`
- `/viewer/KPIs`
- `/viewer/KPIs?hub=Hub&group=Group`
- `/viewer/KPI?hub=Hub&group=Group&name=Name`
- `/viewer/public/Home`
- `/viewer/public/Dashboards`
- `/viewer/public/Dashboard?hub=Hub&group=Group&name=Name`
- `/viewer/public/KPIs`
- `/viewer/user/Home`
- `/viewer/user/Dashboards`
- `/viewer/user/Dashboard?hub=Hub&group=Group&name=Name`
- `/viewer/user/KPIs`



## Configuring Public Access

If a *guest account* exists on the Datazen server, accessible dashboards and kpis accessible can be viewed at */viewer/public* (e.g. *http://server/viewer/public*).

If a URL linking directly to a dashboard's HTML5 view is shared, the *guest* account will be used, if available, to access it when no credentials are provided. Therefore to easily share a dashboard with a large audience:

1. Make it accessible to the *guest* user,
2. Get its URL from the */viewer/public* view,
3. Email it to a list of recipients.

If a dashboard is not accessible by the *guest* user, it can still be shared but only to users who have permission to access it and are logged into the Web Viewer. To link directly to a dashboard, the URL pattern is *http://local.server/viewer/dashboard?hub=HubName&group=GroupName&name=DashboardName*.

# Integrated Windows Authentication

Users can be automatically logged in to the Web Viewer with their Windows credentials (single sign-on) using [Integrated Windows Authentication](#).

To enable this feature:

1. Ensure that Datazen Enterprise Server is configured to use [Active Directory authentication](#).
2. Configure the Datazen Enterprise Server Web API IIS application pool (Datazen.Server.WebApi) identity to run as a domain user in the same domain as the users to be authenticated.
3. Enable both *Anonymous Authentication* and *Windows Authentication* on the Datazen Enterprise Server Web Applications web site (Datazen.Server.Web). This may require installation of the IIS Windows Authentication feature through the Windows control panel.

Clients on the network may now access `http://[server]/viewer/user` to view dashboards. If clients are configured to include `http://[server]` in the *Intranet* zone under their local Internet Explorer settings, they will pass through automatically without a login prompt.

To link directly to a specific dashboard, the URL pattern is `http://[server]/viewer/user/dashboard?hub=[HubName]&group=[GroupName]&name=[DashboardName]`. Note that this URL will only be accessible by others if they are authenticated on the domain and have access to the dashboard via their own Datazen account.

# Embedding Datazen Content into Custom Apps

Dashboards and KPIs which are publicly accessible, or are in an environment which supports [Integrated Windows authentication](#) can be embedded into custom web applications. This is typically done by including an IFRAME element with a link to the desired Datazen content.

## Index views

To produce index views with tiles for each KPI and dashboard, use the url format:

Public access: `[servername]/viewer/public/home`

User Content: `[servername]/viewer/user/home` (via Integrated Windows authentication)

Optional url parameters:

Parameter	Values	Meaning
active	kpis/dashboards	Which section to display by default
group	group name	Only show content from the KPI or dashboard group with the specified name
header	true/false	Whether to display the navigation header at the top
hub	hub name	Only show content from the given hub
name	item name	Only show the KPI or dashboard with the specified name
bg	colour (hex, no hash)	The colour to use for the screen background

## Dashboards

To directly include an interactive dashboard, use the url format:

Public access: `[servername]/viewer/public/dashboard?name=[dashboard name]`

User Content: `[servername]/viewer/user/dashboard?name=[dashboard name]` (via Integrated Windows authentication)

Optional url parameters:

Parameter	Values	Meaning
group	group name	The name of the dashboard group in which the dashboard is found (optional)
hub	hub name	The name of the hub in which the dashboard is found (optional)

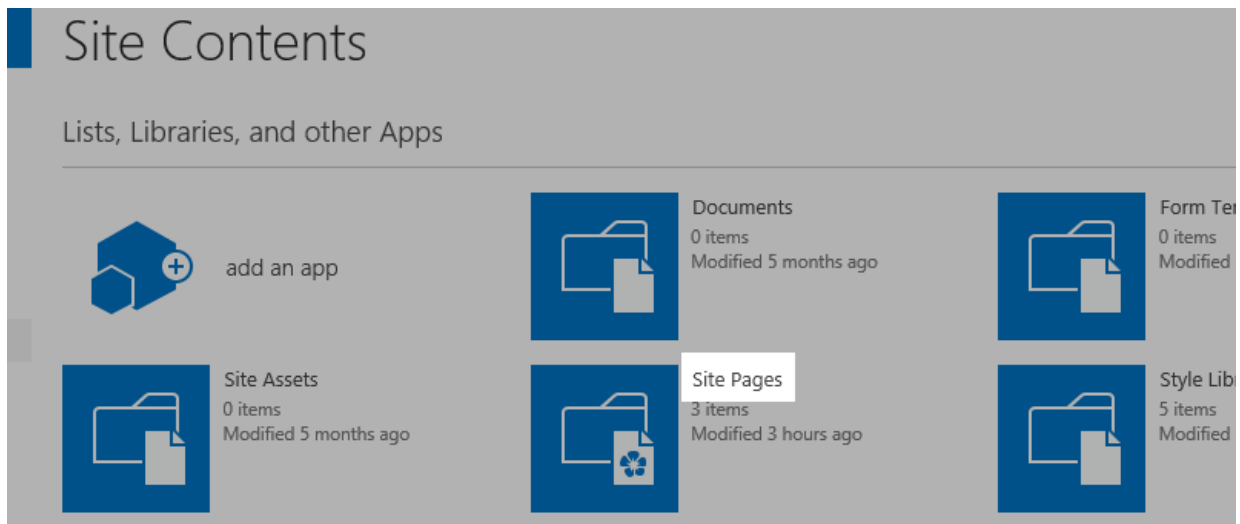
# Embedding Datazen Content Into SharePoint

Dashboards and KPIs which are publicly accessible, or are in an environment which supports [Integrated Windows authentication](#) can be embedded into a SharePoint page. This is typically done by including an IFRAME element with a link to the desired Datazen content.

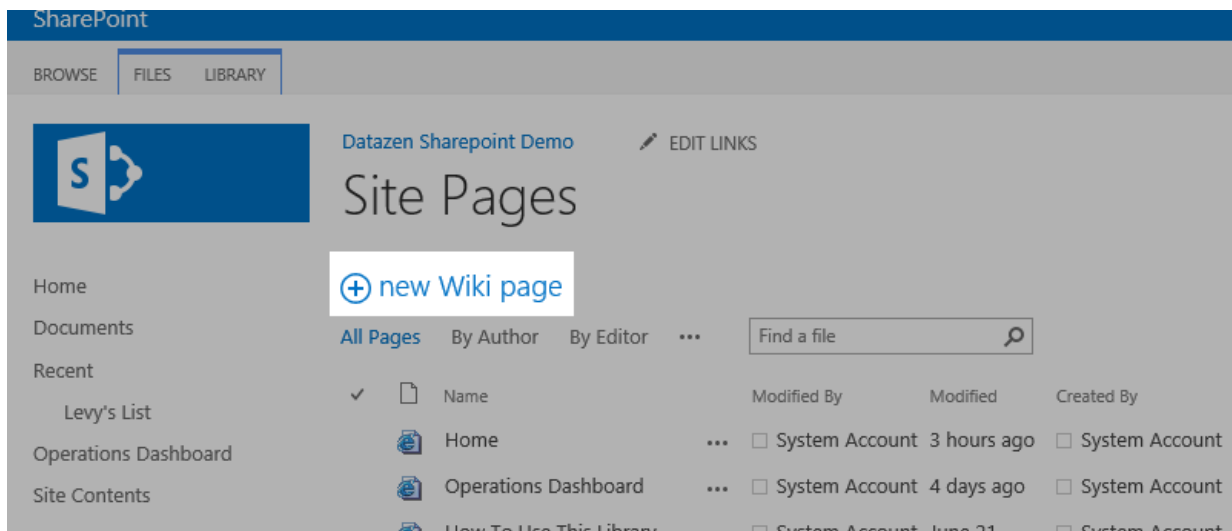
1. Click *Site Contents*.



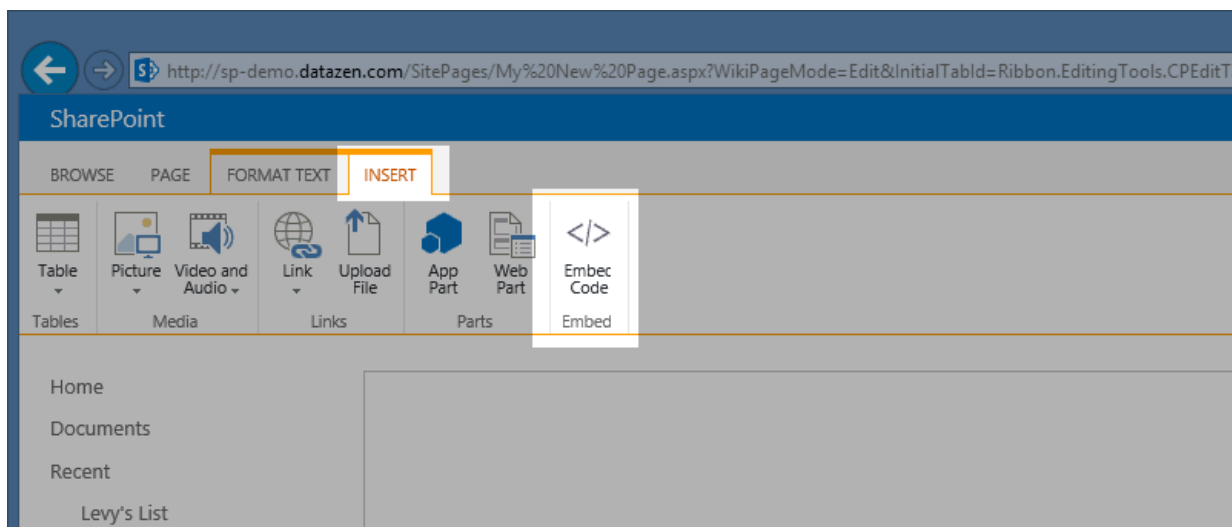
2. Click *Site Pages*.



3. Click *New Wiki Page*.



4. Give the page a name.
5. Click *Create*.
6. Click the *Insert* tab and then *Embed Code*.



7. In the embed code add an IFRAME of the desired size with the appropriate SRC attribute. For information on the format of the SRC url, see [Embedding Content into Custom Apps](#).
8. Click *Insert*.



9. Save the page.



# Server Administration

This section contains the following documents:

- [Windows Services, IIS Sites and Application Pools](#)
- Configuration Settings
  - [Core Service Settings](#)
  - [Data Acquisition Service Settings](#)
  - [Rendering Service Settings](#)
  - [Control Panel Settings](#)
  - [Web API Configuration](#)
- [Batch Creating Users](#)
- [Server Logs](#)
- [Email Templates](#)
- [Repository Backup and Restore](#)
- [Repository Maintenance](#)

# Windows Services, IIS Sites, and Application Pools

## Datazen Core Service

The Datazen Core Service is a Windows service that is responsible for storing dashboards, data and security information in the Repository. This service is a critical component of the Datazen Enterprise Server and the environment will not function if the Core Service is not running.

When Datazen Enterprise Server is first installed, the Windows service *datazen* is created and configured as per the options chosen in the [installer](#). It is possible to [configure the service settings](#) after the Core Service has been installed provided the *datazen* service is not running.

## Datazen Data Acquisition Service

The Datazen Data Acquisition Service is a Windows service (*datazendata*) that periodically queries external data sources and caches results in the Core Service Repository. It is possible to [configure the service settings](#) after the Data Acquisition Service has been installed provided the *datazendata* service is not running.

## Datazen Rendering Service

The Datazen Rendering Service is a Windows service (*datazenrenderingservice*) that generates thumbnails for newly created or updated dashboards.

Configuration for the Rendering Service is located in the [Rendering Service configuration file](#).

## Datazen Control Panel Web Application

The Datazen Control Panel Web Application is an administrative web site for managing various aspects of the Datazen Enterprise Server deployment.

The Control Panel is accessed at the endpoint */cp* of the Web Applications IIS web site *Datazen.Server.Web* and runs in the *Datazen.Server.ControlPanel* application pool.

Configuration for the Control Panel is located in the [Datazen Control Panel web.config file](#).

## Datazen Web Viewer Web Application

The Datazen Web Viewer Web Application is a web site for viewing dashboards and kpis in a web browser.

The Web Viewer is accessed at the endpoint */viewer* (or redirected from the root */*) of the Web Applications IIS web site *Datazen.Server.Web* and runs in the *Datazen.Server.WebViewer* application pool.

Configuration of the Web Viewer is located in the [Datazen Web Viewer web.config file](#).

## Datazen Web API Web Application

The Datazen Web API Web Application is a REST API to which the various Datazen clients talk to.

The Web API is accessed at the endpoint */api3* of the Web Applications IIS web site *Datazen.Server.Web* and runs in the *Datazen.Server.WebApi* application pool.

Configuration of the Web API is located in the [Datazen Web API web.config file](#).

# Configuration Settings

- [Core Service Settings](#)
- [Data Acquisition Service Settings](#)
- [Rendering Service Settings](#)
- [Data Acquisition Service Settings](#)
- [Control Panel Settings](#)
- [Web API Configuration](#)
- [Web Viewer Configuration](#)

# Core Service Settings

The Datazen Core Service configuration settings are set during the [installation procedure](#). The configuration may be modified at any time by stopping the *datazen* service and editing the service's *.config* file located in the *\service* sub-folder of the installation folder.

Some of these settings may be configured in the [Control Panel](#).

## Configuration settings

### **datapath**

The location of the database folder. If this value is preceded with a tilde (~) the location is relative to the *\service* sub-folder of the installation folder. This value should not be modified.

Location:

```
<configuration>
  <appSettings>
    <add key="datapath" value="[path]" />
```

Default value: ~\Data

### **dataport**

The TCP port used for the data api.

Location:

```
<configuration>
  <appSettings>
    <add key="dataport" value="[port]" />
```

Default value: 28952

### **dataport\_restrict**

Restrict access to the core service data port to distributed components. This value should not be modified.

Location:

```
<configuration>
  <appSettings>
    <add key="dataport_restrict" value="[true|false]" />
```

Default value: true

### **apiport**

The TCP port used for the management api.

Location:

```
<configuration>
  <appSettings>
    <add key="apiport" value="[port]" />
```

Default value: 28953

## **apiport\_cert\_thumbprint**

The thumbprint of the SSL certificate to use for distributed components to securely communicate with the core service management api. This certificate is generated automatically during initial product installation.

Location:

```
<configuration>
  <appSettings>
    <add key="apiport_cert_thumbprint" value="[certificate thumbprint]" />
```

## **authtimeout**

The duration (in seconds) to keep remote clients logged in before they must re-authenticate.

Location:

```
<configuration>
  <appSettings>
    <add key="authtimeout" value="[value]" />
```

Default value: 900

## **authtype**

The [authentication mode](#) to use for servicing Datazen Enterprise Server login requests. May have one of the following values:

- default - users are authenticated using their Datazen Enterprise Server credentials
- ad - users are authenticated using Active Directory.
- adfs - users are authenticated using an existing ADFS server.
- external - users are authenticated using an existing external mechanism.

Location:

```
<configuration>
  <appSettings>
    <add key="authtype" value="[default|ad|adfs|external]" />
```

Default value: default

## ad\_server

In [Active Directory authentication mode](#), the server name and port number of the LDAP service. Typically this will be the host name of a domain controller, e.g. `dc1.corp.myorg.com:389`.

Location:

```
<configuration>
  <appSettings>
    <add key="ad_server" value="[server:port]" />
```

This setting may be configured in the [Control Panel](#).

## ad\_domain

In [Active Directory authentication mode](#), the fully qualified domain name, e.g. `corp.myorg.com`.

Location:

```
<configuration>
  <appSettings>
    <add key="ad_domain" value="[value]" />
```

This setting may be configured in the [Control Panel](#).

## ad\_username

In [Active Directory authentication mode](#), the username of an account that will perform the AD query.

Location:

```
<configuration>
  <appSettings>
    <add key="ad_username" value="[username]" />
```

This setting may be configured in the [Control Panel](#).

## ad\_password

In [Active Directory authentication mode](#), the password of the account specified in `ad_username`.

The `ad_password` property is **not** set in the configuration file like the other settings. To set `ad_password` you must run the `Datazen.Server.Service.exe` in console mode:

1. Open a command window (as administrator).
2. Stop the `datazen` service: `net stop datazen`

3. Change directory to the \service sub-folder of the installation folder: `cd "c:\Program Files\Datazen Enterprise Server\service"` for example.
4. Issue the following command: `Datazen.Server.Service.exe /console /ad_password=[thepassword] /shutdown` where *[thepassword]* is the ad\_user password.
5. Start the *datazen* service: `net start datazen`

This setting may be configured in the [Control Panel](#).

## external\_auth\_type

In [external authentication mode](#) whether to locate the authenticated user's Datazen username in the incoming HTTP header or cookie dictionary.

Location:

```
<configuration>
  <appSettings>
    <add key="external_auth_type" value="[header|cookie]" />
```

This setting may be configured in the [Control Panel](#).

## external\_auth\_key

In [external authentication mode](#) the dictionary key (of the container configured in **external\_auth\_type**) whose value is the authenticated user's Datazen username.

Location:

```
<configuration>
  <appSettings>
    <add key="external_auth_key" value="[key]" />
```

This setting may be configured in the [Control Panel](#).

## badloginthreshold

The number of consecutive password failures allowed within the configured **badloginwindow** time span before the account is automatically locked.

Setting this to 0 disables the auto-lockout feature.

Location:

```
<configuration>
  <appSettings>
    <add key="badloginthreshold" value="[number of login failures]" />
```

Default value: 5



## badloginwindow

The time span (in minutes) for tracking consecutive password failures. See **badloginthreshold**.

Location:

```
<configuration>
  <appSettings>
    <add key="badloginwindow" value="[minutes]" />
```

Default value: 5

## can\_hub\_owners\_create\_users

When set to 'false', the server will run in *Enterprise Mode* and hub owners cannot directly add users; they may only invite them. When set to 'true', the server will run in *Demo Mode* and hub owners can add users to a hub.

Location:

```
<configuration>
  <appSettings>
    <add key="can_hub_owners_create_users" value="[true|false]" />
```

## dataviews.updatetimeout

The duration (in seconds) to successfully update (pre-fetch) a data view before aborting.

Location:

```
<configuration>
  <appSettings>
    <add key="dataviews.updatetimeout" value="[value]" />
```

Default value: 60

**Note:** This value can be set per connection type for the following connections types: SQL, SSAS, SharePoint, SQL Azure, OData, Soap, ODBC, and Ole DB:

```
<configuration>
  <appSettings>
    <add key="dataviews.[sql|ssas|sharepoint|sqlazure|odata|soap|odbc|oledb].updatetimeout" value="[value]" />
```

## dataviews.realtimeout

The duration (in seconds) to successfully fetch a data view in real-time before aborting.

Location:

```
<configuration>
  <appSettings>
    <add key="dataviews.realtime timeout" value="[value]" />
```

Default value: 30

**Note:** This value can be set per connection type for the following connections types: SQL, SSAS, SharePoint, SQL Azure, OData, Soap, ODBC, and Ole DB:

```
<configuration>
  <appSettings>
    <add key="dataviews.[sql|ssas|sharepoint|sqlazure|odata|soap|odbc|oledb].realtime timeout" value="[value]" />
```

## dataviews.locktimeout

The duration (in seconds) that data view locks can persist for before being removed. When the Data Acquisition Service updates a data view, it first gets a lock to ensure other instances will not also try to update the view.

Location:

```
<configuration>
  <appSettings>
    <add key="dataviews.locktimeout" value="[value]" />
```

Default value: 3600

## dataviews.maxrecords

The maximum number of records that for a data view. If the number of records returned from a data source exceeds this value, the Data Acquisition Service (for pre-fetched data views) and the Core Service (for real-time data views) will truncate the number of records to this value.

Location:

```
<configuration>
  <appSettings>
    <add key="dataviews.maxrecords" value="[value]" />
```

Default value: 100000

# Data Acquisition Service Settings

The Datazen [Data Acquisition Service](#) configuration settings may be modified at any time by stopping the *datazendata* service and editing the *Datazen.Server.DataService.config* file located in the *\data* sub-folder of the installation folder.

## Configuration settings

### Datazen Enterprise Server connectionString

The location of the REST service (data) of the Datazen Enterprise Server. This setting should not be modified unless the address or port of the Datazen Enterprise Server Core Service machine is changed.

Location:

```
<configuration>
  <connectionStrings>
    <add name="datazenserver" connectionString="[value]"/>
```

Default value: Url=http://localhost:28952

### maxconcurrentupdates

The maximum number of updates to process concurrently. A value of -1 will be mapped to the number of processors/cores in the machine. Personalized data views are processed in sequence as a single update.

Location:

```
<configuration>
  <appSettings>
    <add key="maxconcurrentupdates" value="[value]" />
```

Default value: -1

### sleepduration

A value in milliseconds indicating how long the Data Acquisition Service will sleep for before polling the database for a new set of updates.

Location:

```
<configuration>
  <appSettings>
    <add key="sleepduration" value="[value]" />
```

Default value: 60000 ms

### waitduration

A value in milliseconds indicating how long the Data Acquisition Service will wait for before attempting to schedule an update once maximum concurrency is reached.

Location:

```
<configuration>
  <appSettings>
    <add key="waitduration" value="[value]" />
```

Default value: 250 ms

## Note

Some settings affecting the Data Acquisition Service are configurable in [Core Service settings](#).

# Datazen Rendering Service Settings

The Datazen Rendering Service configuration file is located in the `\renderer\svc` sub-folder of the installation folder. It provides the following configurable settings:

## Datazen Core Service location

The location of the Datazen Server Core Service. This setting should not be modified unless the address or port of the Datazen Server Core Service machine is changed.

Location:

```
<configuration>
  <connectionStrings>
    <add name="datazenserver" connectionString="[value]" />
```

Default value: `Url=http://datazenserver:28952`

## ProcessPoolSize

Number of *DashboardViewer* processes to maintain. Should be set to logical CPU count - 1.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="ProcessPoolSize" serializeAs="String">
        <value>[int]</value>
```

Default value: 1

## ProcessTimeout

Time afforded to *DashboardViewer* to complete a request.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="ProcessTimeout" serializeAs="String">
        <value>[hh:mm:ss]</value>
```

Default value: `00:00:35`

## KillZombiesAtStart

Kill all *DashboardViewer* processes on service start.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="KillZombiesAtStart " serializeAs="String">
        <value>[true|false]</value>
```

Default value: true

## **ProcessKillMemoryLimit**

Size (in Mb) at which a *DashboardViewer* process will be killed and replaced.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="ProcessKillMemoryLimit " serializeAs="String">
        <value>[int]</value>
```

Default value: 984

## **ResponseMemoryFileSize**

Size (in Kb) for the response MemoryMappedFile (images and html).

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="ResponseMemoryFileSize " serializeAs="String">
        <value>[int]</value>
```

Default value: 2048

## **DashboardViewerProcessName**

System name for the *DashboardViewer* process.

Location:

```
<configuration>
  <applicationSettings>
```

```
<Datazen.Rendering.Service.Properties.Settings>
  <setting name="DashboardViewerProcessName " serializeAs="String">
    <value>[value]</value>
```

Default value: DashboardViewer

## SocketPort

The UDP port the service listens on for incoming requests from the Rendering Web Application(s).

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="SocketPort " serializeAs="String">
        <value>[int]</value>
```

Default value: 9000

## SocketBufferSize

The size (in bytes) of the socket buffer.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="SocketBufferSize " serializeAs="String">
        <value>[int]</value>
```

Default value: 1048576

## SocketMaxConnections

Maximum number of simultaneous connections the service supports.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="SocketMaxConnections " serializeAs="String">
        <value>[int]</value>
```

Default value: 1000

## RequestMemoryFileSize

Size (in Kb) for the request MemoryMappedFile.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="RequestMemoryFileSize " serializeAs="String">
        <value>[int]</value>
```

Default value: 16192

## DashboardViewerPath

Absolute path to the *DashboardViewer* executable.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="DashboardViewerPath " serializeAs="String">
        <value>[path to viewer executable]</value>
```

Default value: set by installer

## ServiceTimerInterval

Milliseconds interval to check the Datazen Server Core Service for requests.

Location:

```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="ServiceTimerInterval " serializeAs="String">
        <value>[int]</value>
```

Default value: 3

## ConsoleTimerInterval

Interval to check the request queue for requests.

Location:



```
<configuration>
  <applicationSettings>
    <Datazen.Rendering.Service.Properties.Settings>
      <setting name="ConsoleTimerInterval" serializeAs="String">
        <value>[int]</value>
```

Default value: 3

## Datazen Rendering Web Application Settings

The Datazen Rendering Web Application configuration file is located in the */renderer/web* sub-folder of the installation.

### renderer

Address of a Datazen.Rendering.Service. There can be multiple renderer values defined, each must start with the word *renderer*. Requests are distributed in a round-robin fashion.

Changing these values in a running server causes the collection to be refreshed.

Location:

```
<configuration>
  <appSettings>
    <add key="renderer[id]" value="[address:port]" />
    ...
```

Default value: 127.0.0.1:9000

Example:

```
<add key="renderer01" value="127.0.0.1:9000" />
<add key="renderer02" value="render02.somedomain.com:9000" />
<!-- if both defined, these two rendering servers will be used in a round-robin fashion -->
```

### nojsimagesize

In the absence of either height or width parameters, use this size as the default for non-javascript enabled clients.

Location:

```
<configuration>
  <appSettings>
    <add key="nojsimagesize" value="[widthxheight]" />
```

Default value: 640x480

### ajaxverb

Verb to use for AJAX loading.

Location:

```
<configuration>
  <appSettings>
    <add key="ajaxverb" value="[GET|POST]" />
```

Default value: POST

## **cache.enabled**

Enables or disables caching. Note that even when caching is disabled, images are still written to cache so that they can be retrieved via the ImageCacheHandler; they are deleted upon retrieval.

Location:

```
<configuration>
  <appSettings>
    <add key="cache.enabled" value="[true|false]" />
```

Default value: false

## **cache.optipng**

When optipng.exe is in the */bin* directory of the Datazen Rendering Web Application then PNG compression is enabled for file-type caching. This setting is the command parameters passed to optipng.exe.

Please refer to the [optipng documentation](#) for more information.

Location:

```
<configuration>
  <appSettings>
    <add key="cache.optipng" value="[optipng parameters]" />
```

Default value: -preserve -silent -o1

## **cache.type**

The cache storage mechanism to use. Currently, only *file* is available.

Location:

```
<configuration>
  <appSettings>
    <add key="cache.type" value="[value]" />
```

Default value: file

## cache.interval

The default cache interval in seconds. The default value when none is specified is 1200 (20 minutes).

Location:

```
<configuration>
  <appSettings>
    <add key="cache.interval" value="[seconds]" />
```

Default value: 604800

## cache.maxsize

The maximum size (in megabytes) of the cache before older (and potentially unexpired) items are removed. A value of 0 will disable unexpired content from being removed as the cache grows. The default value when none is specified is 0 (disabled).

Location:

```
<configuration>
  <appSettings>
    <add key="cache.maxsize" value="[size in megabytes]" />
```

Default value: 750

## cache.virtualpath

The virtual path to the ImageCacheHandler. All images are requested via this path.

Location:

```
<configuration>
  <appSettings>
    <add key="cache.virtualpath" value="[value]" />
```

Default value: vcache

Example:

```
<add key="cache.virtualpath" value="vcache" />
<!-- http://server/vcache/image.png -->
```

## cache.file.path

The path where cached content is saved to disk. The path can be absolute or relative to the Datazen Rendering Web

Application installation location.

Location:

```
<configuration>
  <appSettings>
    <add key="cache.file.path" value="[path to cache folder]" />
```

Default value: cache

Examples:

```
<add key="cache.file.path" value="cache" /> <!-- relative -->
<add key="cache.file.path" value="c:\cache" /> <!-- absolute -->
<add key="cache.file.path" value="\\server\share" /> <!-- remote -->
```

# Control Panel Settings

The Control Panel configuration settings are set during the [installation procedure](#). The configuration may be modified at any time by editing the *web.config* file located in the *\controlpanel* sub-folder of the installation folder.

## Datazen Enterprise Server connectionString

The location of the Datazen Enterprise Server Core Service. This setting should not be modified unless the address or port of the Core Service machine is changed.

Location:

```
<configuration>
  <connectionStrings>
    <add name="datazenserver" connectionString="[value]"/>
```

Default value: Url=http://localhost:28952

## SMTP from address

The sender address for all email notifications sent from the Control Panel.

```
<configuration>
  <system.net>
    <mailSettings>
      <smtp from="[smtp email address]">
```

Default value: no-reply@datazen.com

## SMTP host name/port

The host name (or IP address) and port of the SMTP server to use for sending email notifications from the Control Panel.

```
<configuration>
  <system.net>
    <mailSettings>
      <smtp>
        <network host="[smtp host name]" port="[smtp port]" />
```

Default Value: <network host="localhost" port="25" />

**Note:** Several other [SMTP configuration settings](#) are available.

# Web API Configuration Settings

## Datazen Server connectionString

The location of the Datazen Server Core Service. This setting should not be modified unless the address or port of the Core Service machine is changed.

Location:

```
<configuration>
  <connectionStrings>
    <add name="datazenserver" connectionString="[value]" />
```

Default value: Url=http://localhost:28952

## cache.enabled

Enables or disables caching.

Location:

```
<configuration>
  <appSettings>
    <add key="cache.enabled" value="[true|false]" />
```

Default value: false

## cache.type

The cache mechanism to use. May be one of *memory*, *memcached*, *azure* or *redis*

Location:

```
<configuration>
  <appSettings>
    <add key="cache.type" value="[value]" />
```

Default value: memory

## cache.interval

The default cache interval in seconds.

Location:

```
<configuration>
  <appSettings>
    <add key="cache.interval" value="[seconds]" />
```

---

Default value: 60

For additional cache-specific configuration options, see:

- [Configuring Caching with memcached](#)
- [Configuring Caching with Redis](#)
- [Configuring Caching with Azure Cache](#)

# Server Logs

Many of the server-side components of Datazen Enterprise Server log error, trace and debug events to log files. They can be accessed directly on the server or, for some of them, viewed via the Control Panel.

## Core Service and Web API Logs

Core Service logs are stored under `\service\logs` in the Datazen Enterprise Server installation folder. They record notable events related to the operation of the service and roll over daily.

Web API logs can be found under `\webapi\logs` in the Datazen Enterprise Server installation folder.

Events of *debug* or higher priority can also be viewed by the *admin* user in the [Control Panel](#) under the *Log Viewer* option.

## Rendering Logs

Each element of the rendering system maintains its own set of logs. These can be found at:

- `\renderersvc\logs` for the Rendering Service
- `\renderersvc\viewer\logs` for the Rendering Service's viewer executable
- `\renderersvc\web\logs` for the Rendering Web Application



# Repository Backup and Restore

The Datazen Core Service Repository should be periodically backed up for disaster recovery preparation. Backing up the Repository can be performed while the Datazen service is online. The Control Panel *Backup* page can be used to schedule and perform a backup.

The backup will be created in the specified directory **on the machine running the Core Service**. The folder will be created if it does not already exist. If the path is relative, the backup will be created relative to the Core Service *datapath* folder.

**Important:** The Datazen backup mechanism does not modify permissions on the backup destination folder. The server administrator should implement file-level security to prevent unauthorized access to the backup.

For each backup performed, a sub-folder will be created containing the backup itself. The sub-folder name will be in the format: *datazen-[backup date][seconds since midnight]-[random string]*

Home

Configuration

Authentication

Server Users

Email Templates

Branding

Log Viewer

Backup

Status

Schedule

Backup running: false

Last backup started: --

Last back finished: --

Backup directory (e.g. C:\Datazen\Backups):  

Please enter a folder path

Backup Now

Backup and restore page in Control Panel.

## Performing a backup

1. Enter a directory to write the backup files to.
2. Click *Backup Now*.
3. Click *Backup* when prompted to do so.

A backup request has now been sent to the Core Service. The status of the backup can be monitored on this screen, as well as in the Log Viewer.

## Scheduling a backup

1. Click on the *Schedule* tab.
2. Check the *Scheduled backup enabled* checkbox.
3. Define the schedule for the backup.
4. Enter a directory to store the backup files.
5. Click *Save Settings*.

If you make some changes but want to start over, clicking *Load Settings* will reset the form to its last saved settings.

## Restoring a backup

1. If the Core Service (*datazen*) is currently running, open a command prompt (as Administrator) and issue the command: `net stop datazen`. Wait for the service to stop.
2. Navigate to the Core Service folder (*[installation folder]\service*): `cd c:\Program Files\Datazen Enterprise Server\service`
3. If the Core Service data folder (*[installation folder]\service\Data*) exists, delete it: `rmdir /S /Q Data`
4. Create the Core Service data folder: `mkdir Data`
5. If this is **not** the original Core Service machine that the backup was created on, ensure that the *Datazen.Server.Service.config* file contains the correct [instanceid and db encryption key settings](#) from the original Core Service instance. See [Advanced Installation Scenarios](#) for more information.
6. Issue the restore command: `Raven.Server.exe -restore -src [path to backup] -dest Data` and wait for the message `Esent Restore: Restore Complete`
7. If this is **not** the original Core Service machine that the backup was created on, perform a defragmentation operation by issuing the command: `esentutl /d Data\Data`. Wait for the operation to complete.
8. Make sure the Core Service is correctly restored by issuing the command: `Datazen.Server.Service /console /shutdown`. The service should start and then stop successfully.
9. Start the Core Service: `net start datazen`

**IMPORTANT:** Due to a limitation of the underlying storage mechanism, it is *not* possible to restore from a newer OS environment to an older one. For example, restoring to Windows Server 2008 R2 from a backup made from Windows Server 2012 is not permitted.

# Repository Maintenance

## Compacting

If the number of items (data views, dashboards, etc.) has been reduced over time through deletion, user count reduction, or the transitioning from personalized data views to non-personalized, the Datazen Core Service Repository may contain large amounts of unused space. This space can be reclaimed by performing an offline repository compact operation:

1. If the Core Service (*datazen*) is currently running, open a command prompt (as Administrator) and issue the command: `net stop datazen`. Wait for the service to stop.
2. Navigate to the Core Service folder (*[installation folder]\service*): `cd c:\Program Files\Datazen Enterprise Server\service`
3. Issue the command: `esentutl /d Data\Data`. Wait for the operation to complete. **Note:** This operation may take an extended period of time depending on the number of items in the repository.
4. Start the Core Service: `net start datazen`

## Index reset

On rare occasions it is possible that the Datazen Core Service Repository indexes may become corrupt or non-responsive, leading to incorrect results being returned from some repository queries. This can be remedied by resetting the repository indexes from the command line:

1. If the Core Service (*datazen*) is currently running, open a command prompt (as Administrator) and issue the command: `net stop datazen`. Wait for the service to stop.
2. Navigate to the Core Service folder (*[installation folder]\service*): `cd c:\Program Files\Datazen Enterprise Server\service`
3. Issue the command: `Datazen.Server.Service.exe /console /resetindexes /shutdown`. Wait for the operation to complete. **Note:** This operation may take an extended period of time depending on the number of items in the repository.
4. Start the Core Service: `net start datazen`

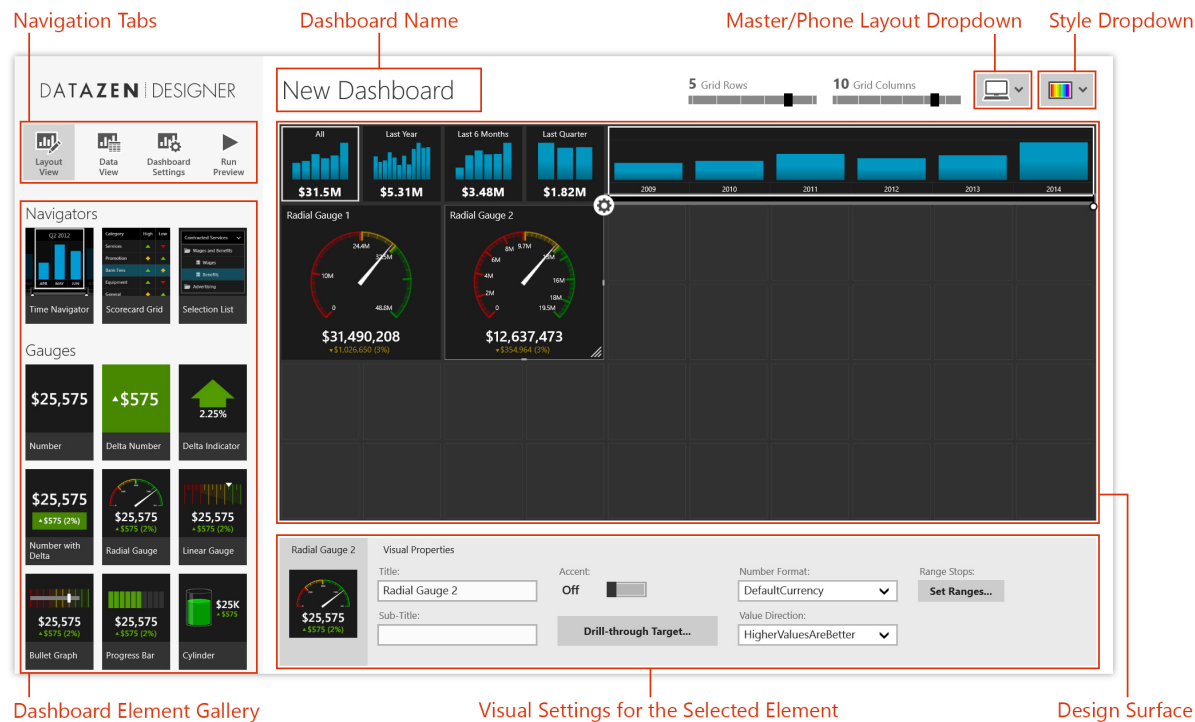
# Creating and Publishing Dashboards

This section contains the following documents:

- [Dashboard Designer](#)
- [Dashboard Runtime](#)
- [Data](#)
- [Configuring Navigators](#)
- [Configuring Visualizations](#)
- [Saving & Publishing Dashboards](#)
- [Advanced Topics](#)

# Dashboard Designer

Datazen Designer enables quick creation of dashboards that scale well to any screen size. The design paradigm is based on a design surface with adjustable grid rows and columns and flexible dashboard elements that adjust perfectly to any amount of real estate. The following diagram shows all the components of the Datazen Designer layout view:



## Data-First vs. Design-First

When creating dashboards, two basic approaches may be taken: data-first, and design-first. Datazen Designer supports both. The data-first approach is to import all required data first and then move next to designing the dashboard and setting data properties on the dashboard elements. This has the advantage of being able to connect each element to real data when it is added to the layout. When using a data-first approach one should be sure that their real data is formatted correctly for use with Datazen.

The design-first approach, on the other hand, is to create a dashboard layout first without importing any data. This is a good way to create a dashboard when you are not sure if the data is formatted correctly. Without real data gallery elements will automatically be bound to generated simulated data, which can be exported and used as a template to describe the required data. The following diagram shows all the components of the Datazen Designer data view:

Dashboard Control Instances

Data Commands

DATAZEN | DESIGNER

Layout View

Data View

Dashboard Settings

Run Preview

Control Instances

Time Navigator 1

Radial Gauge 1

Radial Gauge 2

New Dashboard

Add Data

Refresh All Data

Export All Data

	Date	Metric1	Metric2	Metric3	Metric4	Metric5	Comparison:
1	1/1/2009 12:00:00 AM	305,272.80	121,034.46	248,877.24	185,956.32	122,057.47	282,707.42
2	2/1/2009 12:00:00 AM	239,047.20	96,668.08	191,059.96	145,494.46	96,429.71	282,707.42
3	3/1/2009 12:00:00 AM	250,192.80	100,829.67	196,013.32	153,011.16	99,255.11	282,707.42
4	4/1/2009 12:00:00 AM	267,127.20	107,521.72	213,431.07	162,170.91	107,604.90	274,442.79
5	5/1/2009 12:00:00 AM	297,712.80	117,382.72	244,408.86	182,033.62	120,205.43	274,442.79
6	6/1/2009 12:00:00 AM	337,672.80	136,690.33	277,815.41	202,221.60	133,233.59	274,442.79
7	7/1/2009 12:00:00 AM	417,960.00	166,430.08	332,790.26	252,506.43	167,548.08	342,899.06
8	8/1/2009 12:00:00 AM	393,832.80	153,747.60	317,831.94	238,400.46	157,910.30	342,899.06
9	9/1/2009 12:00:00 AM	335,512.80	131,290.30	269,042.81	199,078.06	132,191.51	342,899.06
10	10/1/2009 12:00:00 AM	301,687.20	122,029.85	247,326.81	181,429.69	121,394.45	320,221.42
11	11/1/2009 12:00:00 AM	251,272.80	103,350.00	201,780.73	151,493.69	99,780.60	320,221.42
12	12/1/2009 12:00:00 AM	275,032.80	107,892.67	222,933.39	159,903.24	109,937.91	320,221.42
13	1/1/2010 12:00:00 AM	476,992.80	188,244.59	372,660.29	285,187.67	193,662.25	371,550.00
14	2/1/2010 12:00:00 AM	222,480.00	90,155.13	177,915.69	134,875.50	88,383.83	371,550.00
15	3/1/2010 12:00:00 AM	267,472.80	105,866.55	208,293.86	159,517.42	107,242.80	371,550.00

SimulatedTable

ExpenseData

SalesData

Radial Gauge 2

Data Properties

Main Value:

SimulatedTable

Metric2

Options...

Comparison Value:

SimulatedTable

Comparison2

Options...

Selected Data View

Data Settings for the Selected Element

Data Views

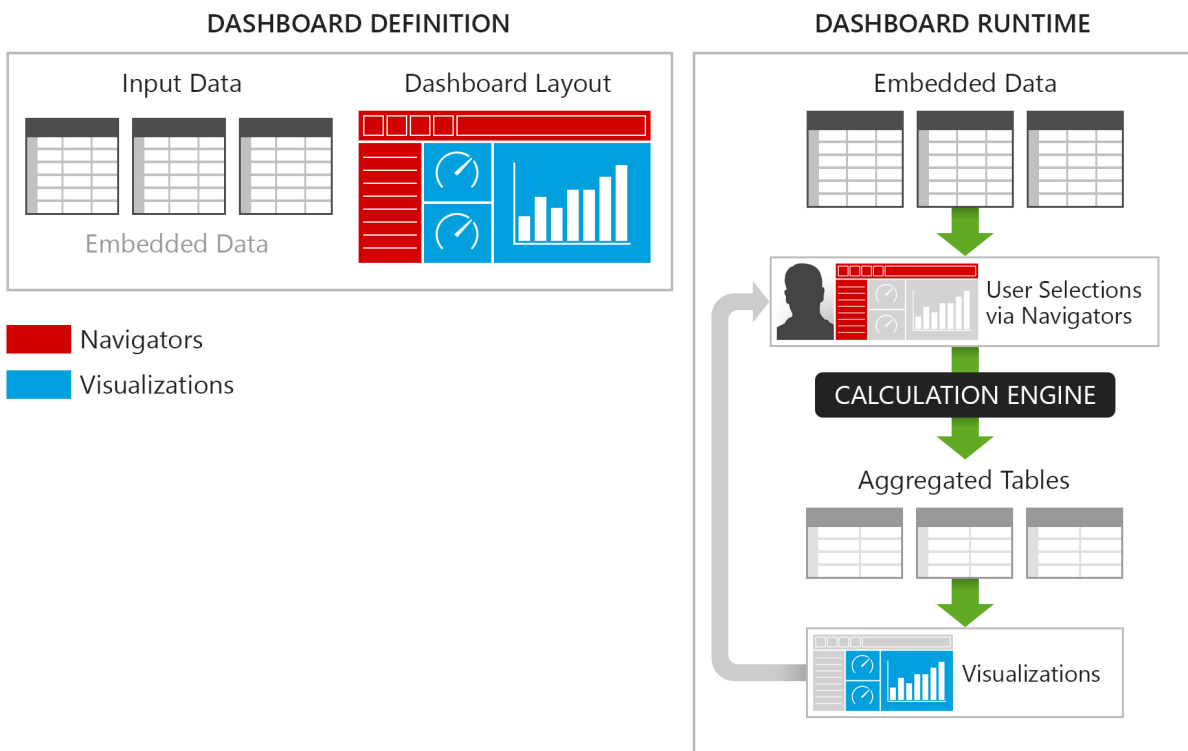
# Dashboard Runtime

The purpose of the Datazen dashboard runtime is to instantiate a dashboard, load the required data and process user actions. Processing user actions – like filter or date/time selections – involves aggregating the original data updating all visual elements of the dashboard. The Datazen dashboard runtime can operate in three different modes:

1. Disconnected mode
2. Connected mode
3. Load on Demand mode

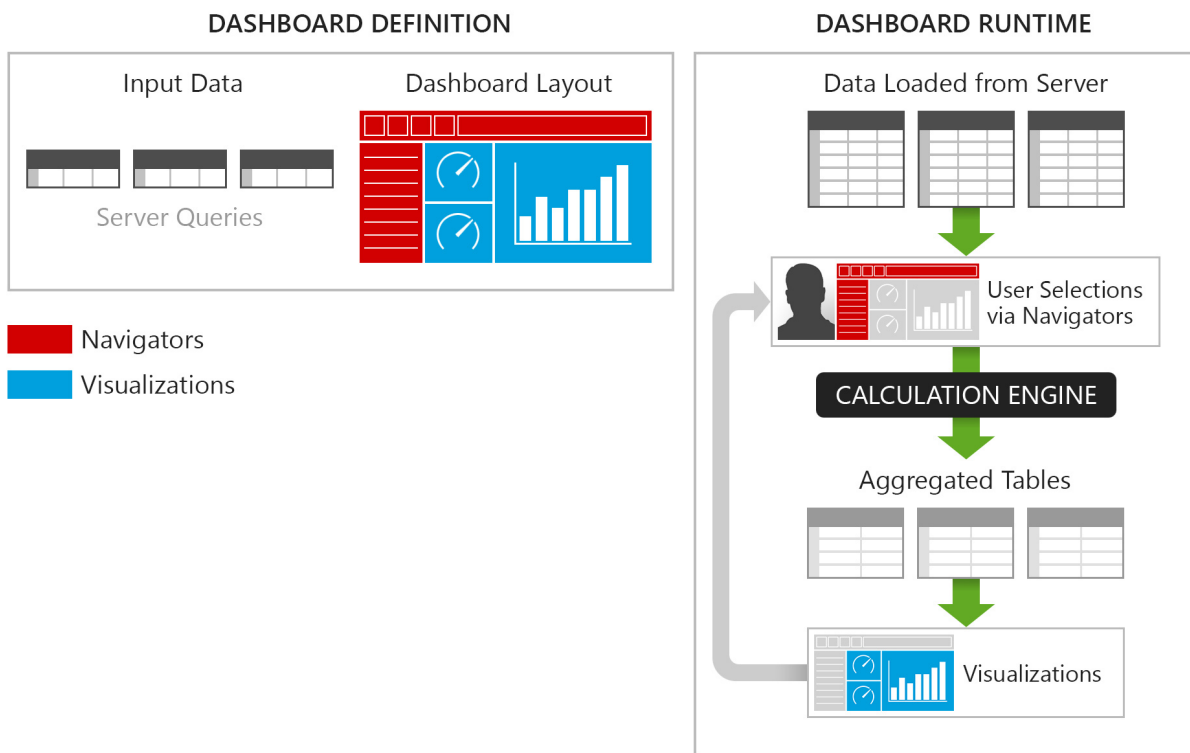
## Disconnected Mode

In the disconnected mode, the device running Datazen is not connected to the server. The device may actually be completely disconnected from the network, or running in “airplane mode”. In this mode, the runtime loads the dashboard definition (containing offline data) instantiates the dashboard and processes user actions by performing filtering and date/time aggregations on the client:



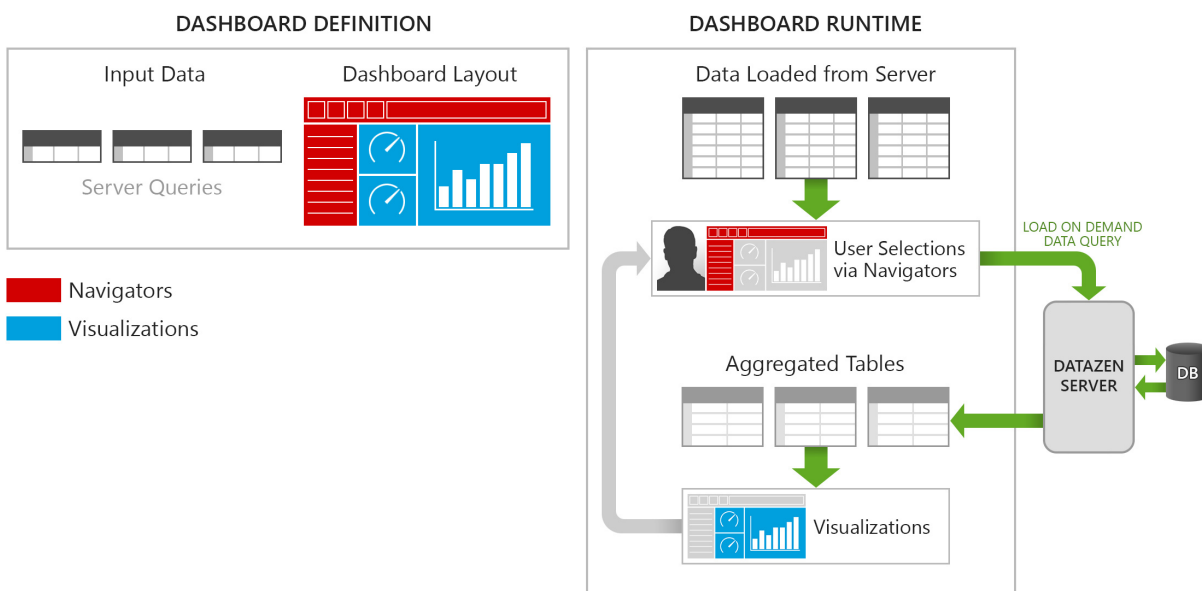
## Connected Mode

In the connected mode, the device running Datazen is connected to the Datazen Enterprise Server. After instantiating a dashboard, the runtime loads fresh data views from the server. From that point on, all user actions are handled on the client, same as in the disconnected mode.



## Load on Demand Mode

The load on demand mode is appropriate when working with very large datasets. In this mode, the device running Datazen is connected to the Datazen Enterprise Server. After instantiating a dashboard, the runtime loads fresh data views from the server. Following that, the runtime handles user actions by issuing new requests to the server – passing selection parameters to parameterized data views – and sending the aggregated results to visualization elements on the dashboard.





# Data

- [Data Model](#)
- [Preparing Dashboard Data](#)
- [Working with Excel Data](#)
- [Working with Live Data Sources](#)
- [Working with Simulated Data](#)

# Data Model

Datazen data model is extremely simple. Data is imported into the Designer as a collection of “data views”. Formal relationships between data views are not required. Lookups from one data view to another will work as long as the key values match. Date/time aggregations are handled by the dashboard runtime and they will match between different data views, even if the date/time data granularity is different between the data views.

There are two types of data views that can be imported:

1. **Local Excel data:** The user selects an Excel document and picks which worksheet(s) will be imported. Once imported, the data is stored within the dashboard definition. To refresh the data from the original Excel file, the Refresh Data command must be used in the designer data view;
2. **Datazen Enterprise Server data:** The user browses the list of data views that have been published on the server and selects the ones to be added to the dashboard. Supported data types include: SQL queries, MDX queries, SharePoint lists, web services, generic ODBC data sources and others. Dashboards published based on server data will always stay connected to the original server data view(s) and reflect the latest state of the data on the server.

Once imported into the Designer, data views are data type agnostic: they are treated equally regardless of their origin. The rest of the dashboard creation and design experience is the same regardless of where the data came from.

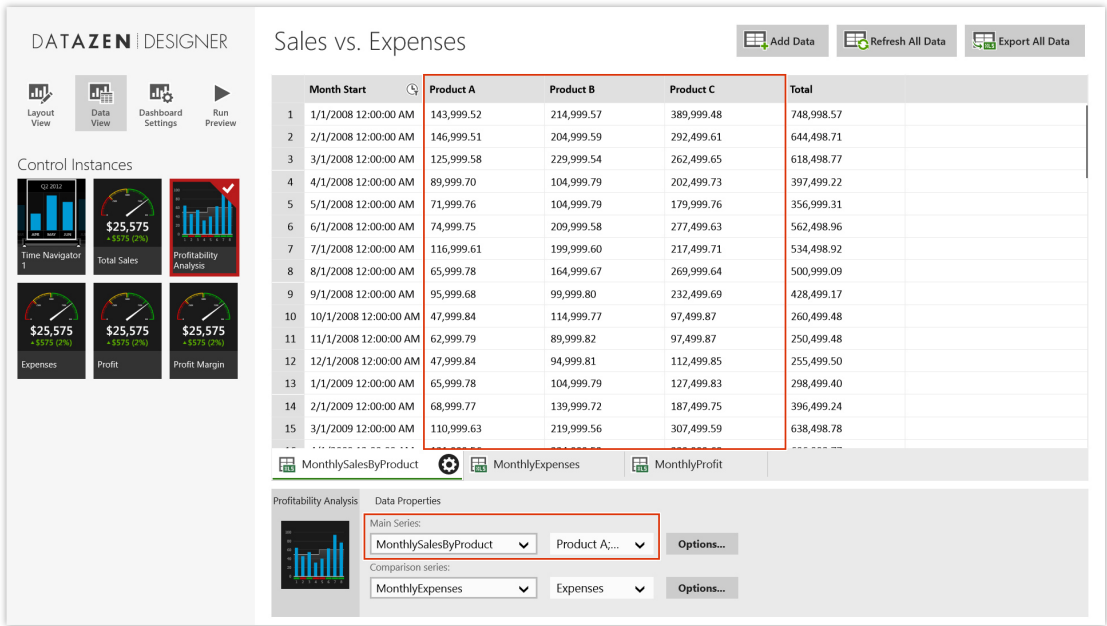
## Connecting Dashboard Elements to Data

Each Datazen dashboard element contains one or more data settings. For example, the Radial Gauge element contains two data settings: Main Value and Comparison Value. Each of these settings points to exactly one field (column) in a specific data view:

The screenshot displays the Datazen Designer interface for a 'Corporate Scorecard' dashboard. On the left, there's a sidebar with 'Control Instances' showing various charts like 'Corporate Structure', 'Human Capital', 'Incidents per Product Line', and three 'Revenue' gauges. The main area shows a table with 12 rows of data across 7 columns: Division, Revenue, Revenue Target, Expenses, Expense Projections, Profit, and Target Profit. Below the table, there's a 'Data Properties' section for a selected 'Revenue' gauge. It shows 'Main Value' set to 'PerformanceByDivision' and 'Revenue', and 'Comparison Value' set to 'PerformanceByDivision' and 'Revenue Targ'. The 'Revenue' gauge shows a value of \$25,575 with a 2% increase.

Division	Revenue	Revenue Target	Expenses	Expense Projections	Profit	Target Profit
1 Division A1	30,527,280.00	22,967,145.87	23,849,640.00	15,907,867.00	6,677,640.00	7,059,278.87
2 Division A2	23,904,720.00	22,967,145.87	11,124,000.00	15,907,867.00	12,780,720.00	7,059,278.87
3 Division A3	25,019,280.00	22,967,145.87	13,373,640.00	15,907,867.00	11,645,640.00	7,059,278.87
4 Division A4	26,712,720.00	27,441,243.12	20,285,640.00	14,848,752.00	6,427,080.00	12,592,491.12
5 Division B1	29,771,280.00	27,441,243.12	16,470,000.00	14,848,752.00	13,301,280.00	12,592,491.12
6 Division B2	33,767,280.00	27,441,243.12	15,587,640.00	14,848,752.00	18,179,640.00	12,592,491.12
7 Division B3	41,796,000.00	42,178,783.20	19,386,000.00	13,889,247.00	22,410,000.00	28,289,536.20
8 Division B4	39,383,280.00	42,178,783.20	16,794,000.00	13,889,247.00	22,589,280.00	28,289,536.20
9 Division C1	33,551,280.00	42,178,783.20	11,807,640.00	13,889,247.00	21,743,640.00	28,289,536.20
10 Division C2	30,168,720.00	29,137,151.94	12,924,360.00	19,615,238.00	17,244,360.00	9,521,913.94
11 Division C3	25,127,280.00	29,137,151.94	23,400,360.00	19,615,238.00	1,726,920.00	9,521,913.94
12 Division C4	27,503,280.00	29,137,151.94	20,268,360.00	19,615,238.00	7,234,920.00	9,521,913.94

The dashboard runtime provides aggregated values for the gauge, based on user selections. Note that the Comparison Value of the same Radial Gauge instance can be bound to a field from a different data view. The following example shows how multiple chart series can be bound to a collection of fields in a data view:



# Preparing Dashboard Data

Datazen supports a number of complex data features including filtering, aggregation, and time slicing; however certain points should be kept in mind while preparing data. Pre-aggregating data can optimize both dashboard creation and use and is, on occasion, required by some dashboard designs.

For best results, queries or documents should be pre-aggregated to 10,000 records or fewer. For record sets larger than 10,000 records configuring load-on-demand is suggested. See [Data Access](#) for information on when load-on-demand and parameterized data views and dashboards should be used.

When dealing with date and time intervals for use in a dashboard, particularly with the TimeNavigator, it's important that the date/time column is properly formatted so that it can be identified as such. The following are examples of valid date/time formats:

```
05/01/2009
2009-05-01
05/01/2009 14:57:32.8
2009-05-01 14:57:32.8
2009-05-01T14:57:32.8375298-04:00
5/01/2008 14:57:32.80 -07:00
1 May 2008 2:57:32.8 PM
Fri, 15 May 2009 20:10:57 GMT
```

Date- and time-based datasets can, in most cases, be described by one or more date/time intervals, such as hourly, daily, monthly, quarterly, and yearly. Datazen is able to combine multiple tables of different granularity and display them on a single dashboard however it's good to keep in mind the relevant intervals from the original dataset(s) as this can help when deciding what date/time filter options to present to the user in the final dashboard.

Datazen is able to filter data based on both date/time fields and key fields. While key fields can be numeric in most cases they will be either an ID or a string value. In order to prepare a filter field for use with a navigator element such as the Selection List the filter key should be contained within a single column of the desired data table. In this way the table rows may be grouped according to the value in the filter column. Having multiple columns contain different filter keys, or filter criteria, allows for dashboards with multiple filter navigators to be used together in a cascading fashion, or individually applied.

Industry	Country	Region
Banks	AFGHANISTAN	ASIA
Commercial & Professional Services	AFGHANISTAN	ASIA
Food, Beverage & Tobacco	AFGHANISTAN	ASIA
Media	AFGHANISTAN	ASIA
Pharmaceuticals	AFGHANISTAN	ASIA
Food & Staples Retailing	ALBANIA	EUROPE

Key-based filters may also be hierarchically structured for use with a Selection List in a tree structure. To prepare data for use in this type of scenario a look-up table should be created describing the hierarchical structure. This is easily accomplished by using a table structure that includes a Key column and a Parent Key column to indicate where a node belongs in the overall hierarchy.

ItemKey	ParentItemKey
Financial	
Industrials	
Consumer Staples	
Consumer Discretionary	
Health Care	
Information Technology	
Banks	Financials
Real Estate	Financials
Diversified Financials	Financials
Insurance	Financials
Commercial & Professional Services	Industrials
Capital Goods	Industrials
Transportation	Industrials
Food, Beverage & Tobacco	Consumer Staples
Food & Staples Retailing	Consumer Staples
Household & Personal Products	Consumer Staples
Media	Consumer Discretionary
Automobiles and Components	Consumer Discretionary
Consumer Durables and Apparel	Consumer Discretionary
Consumer Services	Consumer Discretionary
Retailing	Consumer Discretionary

Pharmaceuticals	Health Care
Health Care Equipment & Services	Health Care
Software & Services	Information Technology
Technology Hardware & Equipment	Information Technology
Telecommunication Services	Information Technology

# Working with Excel Data

Here are some things to keep in mind when preparing an Excel file and worksheets for use with a dashboard:

## DO

- Have one worksheet per data set
- Have column headers in the first row
- Keep data types consistent within each column
- Format cells as proper types in Excel
- When using formulas, ensure that the entire column is calculated using the same formula
- Use Excel 2007 or later
- Save Excel files with the extension XLSX

## DO NOT

- Include images, graphs, pivot tables or other embedded objects in data set worksheets
- Include total or calculated rows
- Keep the file open in Excel when importing
- Format numbers manually by adding currency or other symbols
- Use Excel 2003 or earlier
- Use files with the extension XLS

## Worksheets

When preparing an Excel file for use with a dashboard one should enforce a policy of having one dataset per worksheet. Each individual worksheet will be imported into the dashboard designer as a separate table. Identically named worksheets from multiple Excel sources will be renamed upon importing by appending incrementing numbers, for example if three worksheets entitled "MyWorksheet" are imported the second and third will be renamed to "MyWorksheet0" and "MyWorksheet1". The screenshot below illustrates the first few rows of an ideal Excel worksheet ready for import.

	A	B	C	D	E	F	G
1	Date	Region	Product A	Product B	Product C	Total	
2	1/1/08 12:00 AM	Africa	143999.52	214999.57	389999.48	748998.57	
3	2/1/08 12:00 AM	North America	146999.51	204999.59	292499.61	644498.71	
4	3/1/08 12:00 AM	Asia	125999.58	229999.54	262499.65	618498.77	
5	4/1/08 12:00 AM	North America	89999.7	104999.79	202499.73	397499.22	
6	5/1/08 12:00 AM	Africa	71999.76	104999.79	179999.76	356999.31	
7	6/1/08 12:00 AM	Europe	74999.75	209999.58	277499.63	562498.96	
8	7/1/08 12:00 AM	Caribbean	116999.61	199999.6	217499.71	534498.92	
9	8/1/08 12:00 AM	Asia	65999.78	164999.67	269999.64	500999.09	
10	9/1/08 12:00 AM	Europe	95999.68	99999.8	232499.69	428499.17	
11	10/1/08 12:00 AM	Central America	47999.84	114999.77	97499.87	260499.48	
12	11/1/08 12:00 AM	North America	62999.79	89999.82	97499.87	250499.48	
13	12/1/08 12:00 AM	Europe	47999.84	94999.81	112499.85	255499.5	
14	1/1/09 12:00 AM	North America	65999.78	104999.79	127499.83	298499.4	
15	2/1/09 12:00 AM	Asia	68999.77	139999.72	187499.75	396499.24	
16	3/1/09 12:00 AM	Middle East	110999.63	219999.56	307499.59	638498.78	
17	4/1/09 12:00 AM	North America	131999.56	234999.53	239999.68	606998.77	
18	5/1/09 12:00 AM	North America	140999.53	189999.62	337499.55	668498.7	

Example data in Excel.

As you can see in the example above, it is preferable that the first row contain the name of the metric contained in that

column. Upon adding this worksheet to the designer, these column headers will be preserved for easy reference in gallery element settings. Column headers are not required, however, and if missing will be auto-generated using the Excel A,B,C,...,AA,BB,... convention.

## Cells

It is important that cell data is consistent within each column of a worksheet's dataset. Each column will be assigned a data type upon importing. Data types will be automatically detected as either string, double (numeric), boolean (true/false) or datetime. Mixing data types within the same column of a worksheet can cause this detection to be inaccurate or fail completely. This detection accounts for possible column headers being of string type. Cells should be formatted as the correct type in Excel to ensure that dashboard designer detects the desired types. In the above example the six columns would be typed as a datetime column, a string column, and four double columns.

If a worksheet contains calculated cells or formulas, only the resulting display value will be imported into the dashboard designer.

## Column Headers

Worksheets may, or may not contain headers in the first row of the data. First-row headers are automatically detected when importing Excel worksheets. This is done by comparing the data types of the first two cells in each column. If the data types of the first two cells in any column do not match the first row is determined to be column headers. For this reason it is sometimes important to prefix numeric column headers with a string so that they may be detected as headers and imported into the dashboard designer as such.

## File Location and Refreshing Excel Data

There are no restrictions as to where on your file system you can store Excel files to be imported into the dashboard designer. However, if the file is moved or renamed after importing, it will prevent the ability to refresh that data through the "refresh all data" command found in the Data View. It's important to note that the dashboard designer does not automatically refresh Excel data; this must be done by the dashboard author through the designer refresh command. If you wish to take advantage of this feature, ensure that any imported Excel files reside in a permanent location on your file system.

## Dates

Dates fields are essential to many dashboards thus it is important to ensure that cells are properly formatted as dates in Excel. In some cases this means a conversion is necessary. This section describes some techniques for converting cells from text to dates in Excel.

```
Week 24-2013=DATE(MID(A2,9,4),1,-2)-WEEKDAY(DATE(MID(A2,9,4),1,3))+MID(A2,6,2)*7
```

```
2013/03/21=DATEVALUE(A1)
```

```
2013-mar-12=DATEVALUE(RIGHT(A1,2)&"-"&MID(A1,6,3)&"-"&LEFT(A1,4))
```

Once a conversion like this has been applied the cells must be formatted as dates by selecting them, or the entire column, selecting Format Cells from the Context Menu, and Date from the Category list in the Format Cells dialog. Excel's text-to-columns wizard may also be used to convert text cells to properly formatted dates.



## Unsupported

Worksheet data in formats other than those described above could cause undesired results when imported. It is a good idea to restrict worksheets in an Excel document to only those which are in the correct format for use with a dashboard.

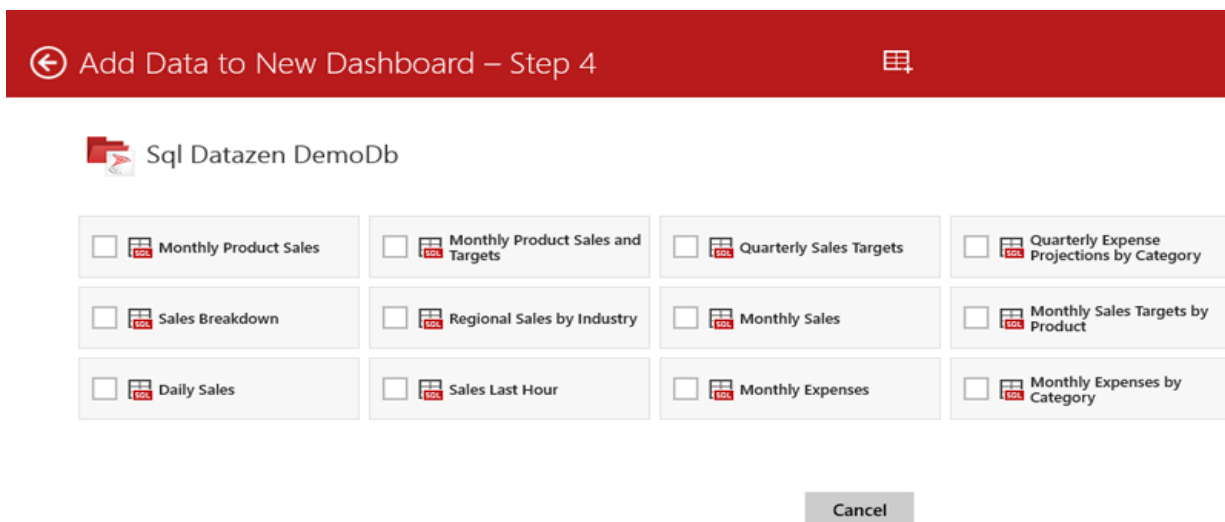
Custom objects embedded into Excel worksheets including pivot tables, visualizations, and images will not be imported into the dashboard designer.

## Working with Live Data Sources

In addition to being able to load data from local [Excel](#) files Datazen can, in conjunction with a Datazen Enterprise Server, also access live data from almost any source. Accessing live data requires a Datazen Enterprise Server; each live data source is configured on that Datazen Enterprise Server through its control panel by an administrator.

Once a connections and data views, or queries, are configured, live data sources are proxied through the Datazen Enterprise Server and made available to Datazen clients for inclusion in dashboards.

After a user has connected to a Datazen Enterprise Server from the client, importing live data views into a dashboard is as easy as selecting Datazen Enterprise Server instead of local excel from the “add data” dialog. Selecting Datazen Enterprise Server will present the user with the data sources available sorted into folders by the server administrator. Selecting a data source will present the user with the available data views in that source, one or more of these views may then be checked off and imported into the dashboard designer.



Live data sources are available from servers.

Thanks to Datazen Enterprise Server, once a live data source has been imported into the dashboard designer the design process may proceed in exactly the same as with local data.

### Real-time data and Parameters



Data sources can be configured to perform real-time data retrieval – every time a dashboard that utilizes real-time data is viewed, the original data store is queried for data.

These real-time sources may use parameters in their query to filter data before it is sent to a dashboard. When a parameterized source is added to a dashboard, the “Parameters” option will become available in the source’s settings rose. Invoking this option will present a dialog that allows you to override the default value for each parameter with either a static value or the property of a navigator on the dashboard.

# Working with Simulated Data

When a gallery element is placed on the design surface, simulated data for that control is immediately generated. This data serves a number of useful purposes when creating dashboards.

	Category	Y	Metric1	Metric2	Metric3	Metric4	Metric5	Comparison1
1	Item A1		145,748.00	241,698.00	560,122.00	3,430.00	759,778.00	118,044.00
2	Item A2		122,024.00	773,750.00	156,859.00	156,615.00	485,458.00	142,272.00
3	Item A3		231,842.00	103,066.00	863,856.00	891,691.00	704,546.00	81,281.00
4	Item A4		991,127.00	565,861.00	332,030.00	997,866.00	641,611.00	866,537.00
5	Item B1		934,386.00	183,375.00	424,880.00	550,112.00	2,992.00	11,422.00
6	Item B2		8,644.00	234,560.00	766,129.00	365,567.00	983,919.00	42,681.00
7	Item B3		674,421.00	650,233.00	231,684.00	487,082.00	399,658.00	556,162.00
8	Item B4		194,827.00	466,473.00	451,987.00	41,587.00	23,109.00	48,467.00
9	Item C1		565,675.00	847,905.00	331,714.00	241,004.00	334,066.00	756,567.00

 SimulatedChartDataGridTable  SimulatedTable

An example of simulated data.

Firstly, simulated data can help immensely when taking a design-based approach to dashboard creation. Initially populating elements with simulated data allows the rapid creation of dashboard prototypes without having to address specific data requirements. These dashboards can then be evaluated for the overall aesthetics and effectiveness.

Secondly, simulated data provides a template which accurately represents the data requirements of a particular dashboard design. By using the Export All Data command located at the top right of the Data View, an Excel document containing the simulated data is generated, allowing for quick substitution of actual data, ready for import.

## Simulated Data Behavior

The simulated data generated is tailored specifically to the dashboard being created. As more elements are placed on the design surface, the associated simulated data will grow and change to provide the best possible experience short of real data. This evolution ensures that extra fields and filter possibilities are available in case the user adds extra series to chart visualizations or expands the scope of one or more dashboard elements in another way.

Simulated data may be exported to an Excel file, as mentioned previously, creating a perfect data template for the associated dashboard. Users may simply swap in their own real data into the Excel file and import it back into the dashboard designer.

Once all controls have been bound to real data, simulated tables which are no longer in use are automatically removed from the dashboard. Simulated tables still referenced by elements on the design surface cannot be removed.

It's important to note that simulated data does not add to the overall dashboard footprint as they are not serialized with the dashboard but generated on-the-fly at runtime.

# Configuring Navigators

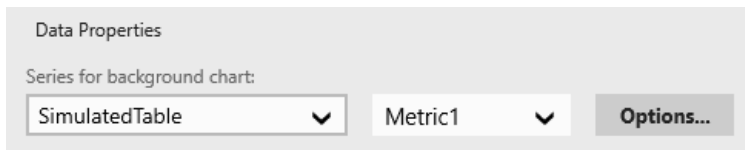
## Navigators & Filtering Concepts

Navigators are used to filter input data tables into smaller subsets. Time-based navigators filter tables by selecting rows that fall into the desired time range. Selection-based navigators filter tables by selecting rows where a particular column value matches the desired key value; or, in cases of hierarchical trees, where a particular column value belongs to the sub-tree of the desired key value.

Navigators can be used to provide filter values in offline datasets, or for load-on-demand and parameterized queries in connected scenarios. For more information on which functionality you should consider please refer to [Data Access](#).

## Time Navigator

As its name implies, the time navigator filter is used to select a range of data bounded by a time range. Any control that is connected to time-based data can opt-in (default) to the time navigator filter. If a table contains more than one time-based column, only the first is used for filtering. The series table drives the embedded visualization and determines the overall date range of the dashboard.

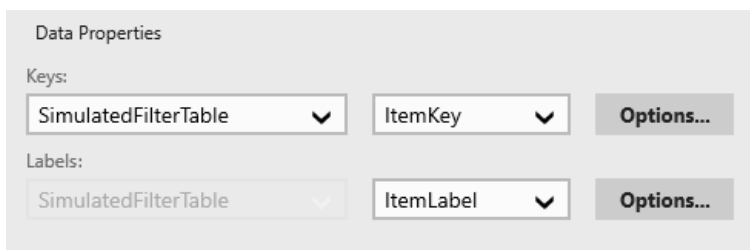


The screenshot shows a 'Data Properties' panel for a Time Navigator. It has a section titled 'Series for background chart:' containing two dropdown menus. The first dropdown is set to 'SimulatedTable' and the second is set to 'Metric1'. To the right of these dropdowns is a button labeled 'Options...'.

Time Navigator series properties.

## Selection List

The selection list filters input data by matching the selected key value of the filter to the value of a specified column for each row of a filtered table. In the data properties panel, first select a table that will provides the keys to be filtered and choose the key column, then select a column for the label that will be displayed. The key column and label column can be the same.



The screenshot shows a 'Data Properties' panel for a Selection List. It has two sections: 'Keys:' and 'Labels:'. The 'Keys:' section has two dropdown menus, the first set to 'SimulatedFilterTable' and the second set to 'ItemKey', followed by an 'Options...' button. The 'Labels:' section has two dropdown menus, the first set to 'SimulatedFilterTable' and the second set to 'ItemLabel', followed by an 'Options...' button.

Selection List data properties.

In the case of hierarchical tree data, a parent key column must be selected.

**Data Properties**

Keys:

SimulatedFilterTable ▼	ItemKey ▼	Options...
------------------------	-----------	------------

Parent Keys:

SimulatedFilterTable ▼	ParentItemKe ▼	Options...
------------------------	----------------	------------



Labels:

SimulatedFilterTable ▼	ItemLabel ▼	Options...
------------------------	-------------	------------

Selection List tree structure data properties.

Once the data properties have been set, tables that are to be filtered must be configured. In the filtered tables panel, check each table to be filtered and select the column to filter by. This column will be used to match values with the selection list's key column.

**Tables Filtered by Selection List 1**

<input type="checkbox"/>	 SimulatedFilterTable	▼
<input checked="" type="checkbox"/>	 SimulatedTable	FilterKey ▼

Selection List filter configuration.

## Scorecard Grid

The scorecard grid filter functions very much like the selection list filter, but in addition, it also displays score indicators and value columns. After selecting the key, label, and optional parent key columns, an input table and column must be selected to provide the scorecard with data. The scorecard data column should be filterable by the key column.

**Data Properties**

Keys:

SimulatedFilterTable ▼	ItemKey ▼	Options...
------------------------	-----------	------------

Labels:

SimulatedFilterTable ▼	ItemLabel ▼	Options...
------------------------	-------------	------------

Data Keys:

SimulatedTable ▼	FilterKey ▼	Options...
------------------	-------------	------------

Scorecard Grid data properties.

To configure the score indicators, switch to the data view and locate the panel in the lower-right corner. From here, a score indicator can be added by clicking the "add score" button.

Scorecard Grid column configuration.

Name the score indicator as desired and configure its properties by clicking the options button.

Scorecard Grid visualization configuration.

- Gauge type: Choose a gauge type from the list; valid gauges types are delta arrow, delta indicator, delta background, delta foreground, and progress bar.
- Value field: Choose the desired value field from the table.
- Comparison field: Choose the desired comparison field from the table.
- Value direction: Choose whether higher values are good or lower values are good.

To add a value indicator, click the “add value” button. Name the value indicator as desired, choose its source column from the table, and select how it will be formatted.

## Configuring Gallery Elements

Gallery elements are configured to use filters by clicking the options button for a particular input in data view. Filters are enabled by toggling the appropriate checkbox.

Filtered by:

☒ Time Navigator 1

☒ Selection List 1

Aggregation:

☒ Sum

☐ Avg

☐ Count

☐ Min

☐ Max

☐ First

☐ Last

Cancel

Done

Options...

Gallery element filter configuration.

## Cascading Filters

Filters can also be cascaded together so that the selected value of one will filter the available values in a second. To cascade filters, apply the filters to the Key column as you would a regular gallery element.

# Configuring Visualizations

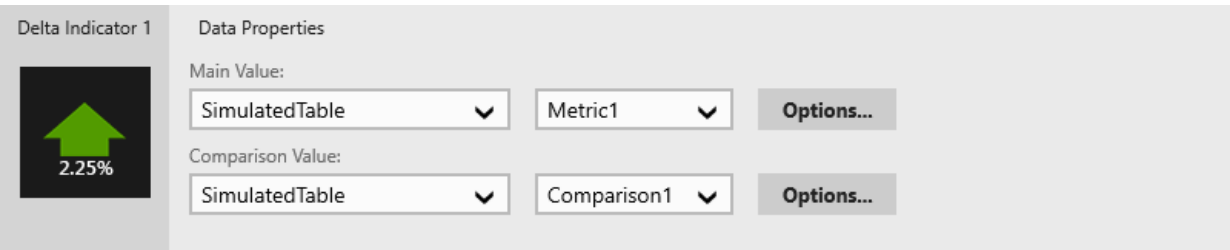
## Gauges



Gauge visualizations.

Gauges are the most basic and most widely used visualization controls in Datazen. All gauges share at least one common property, value. In order for a gauge to function with real data, its value property must be set to a numeric field in one of the data tables available to the dashboard in which it resides. The value property can be set by selecting the gauge visualization on the design surface, switching to the data view and using the drop-down lists in the data properties panel to select a data table and a numeric field.

Some gauges, such as the delta number and delta indicator visualizations, are able to display a comparison value or the delta relationship between the value and comparison value. The comparison value, when supported, may also be set by selecting the gauge on the design surface and using the drop-down lists in the data properties panel to select the desired data table and numeric field which represents the comparison value.



Gauge data properties.

Aggregation options are also available for gauge elements. The default is sum, which will display the total of all values contained within the current filtered data available to the gauge control. When selecting an alternate aggregation method for the value property keep in mind that you may also want to change the aggregation method on the comparison value property, though in some cases mixing aggregation methods may be desired.

If the dashboard has any navigators, a gauge element may be bound to one or more of these by checking off the desired Navigators in the “filtered by” section of the option popup found in the data properties panel. For the sake of versatility, a gauge’s value and comparison value may be bound to one or more different navigators. This powerful feature allows virtually endless options for delta gauge elements.

Along with the data properties which connect gauge elements to data fields, there are a number of functional and visual properties which can be configured to customize an element. The value direction property is one of these and may be set to either HigherValuesAreBetter or LowerValuesAreBetter. This determines whether positive values will be colored green, indicating a desirable change for the better, or red, indicating an undesirable change for the worse. The value



direction property relates only to Gauge elements which support a comparison value and the color of the delta gauge will be determined by the sign of the delta integer and the value direction property setting.

The second gauge-specific non-data property is range stops. Range stops can be adjusted via a popup accessed below the design surface once a gauge element has been selected. This property determines at what percentage of its comparison value visualization should be presented as on-target (green), neutral (amber) and off-target (red) – a gauge’s comparison value being the target.

Another non-data property of gauge element, and one shared by many other elements, is number format. The number format property can be accessed below the design surface when an element is selected. It determines how numbers displayed in the element should be formatted – standard versus currency, for instance. This property is not dashboard-wide and should be set on each element.

## Charts

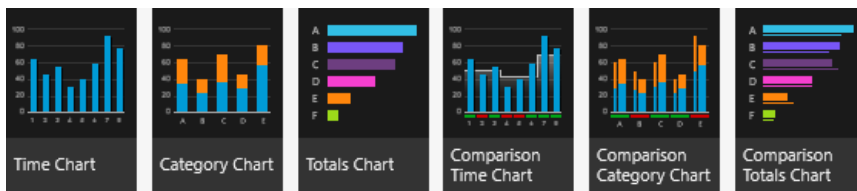


Chart visualizations.

An essential part of dashboard design and data visualization, Datazen offers a variety of chart elements which can be configured to cover a wide range of scenarios. Some properties apply to all chart elements, and others only to specific chart elements.

The number format property can be accessed below the design surface when an element is selected; this property determines how numbers displayed in the chart element should be formatted – standard versus currency for instance. This will apply to axis annotations, as well as data point popups. This property is not dashboard-wide and should be set on each element.

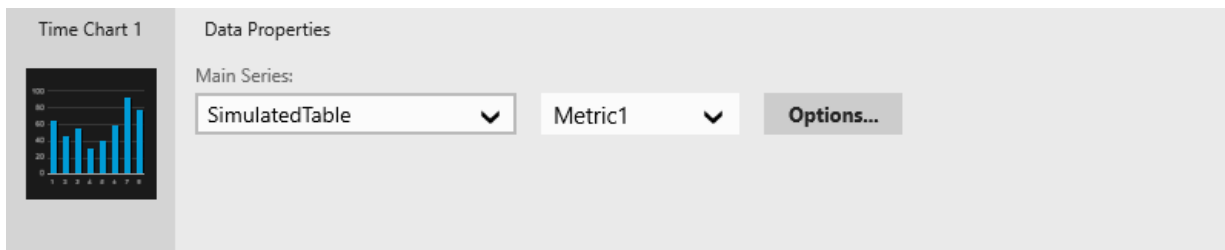
A legend may be enabled for chart elements by toggling the show legend property on the layout view.

Each individual metric, or value, displayed on a chart element is referred to as a series; multiple series can, and do, share both a common x-axis and a common y-axis. Series are defined in the data properties panel of the data view by selecting one or more data tables and fields. Each field will result in an individual series of datapoints on the chart visualization with its own color.

There are three basic types of chart types in Datazen: time, category, and totals. These three chart types have matching comparison chart elements, which allow the chart’s visualization to show two distinct sets of series for comparison.

### Time Chart

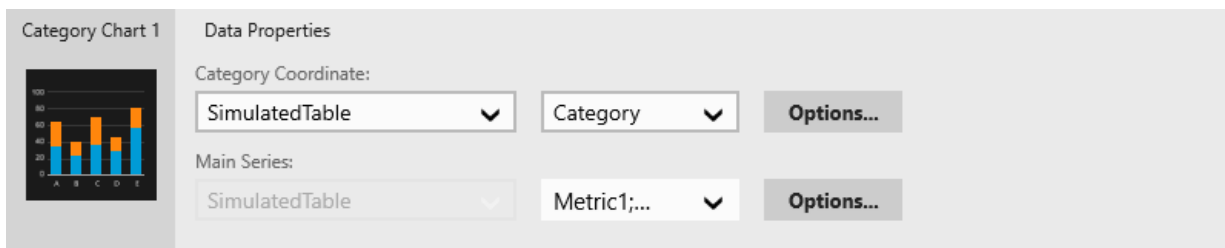
The time chart is the most basic chart element in Datazen and can be configured by setting its main series property to a numeric field in a data table. The time (and date) axis of the chart will automatically be set to the first valid date/time field in the data table.



Time Chart data properties.

## Category Chart

The category chart, unlike the time chart, will allow an x-axis relating to a grouping rather than a date/time field. This grouping is assigned through the category coordinate property accessible in the data properties panel of the data view. Category coordinates are restricted to fields of string type. The data will be grouped by this field and displayed on the resulting chart with the category coordinate field comprising the x-axis. The main series of category charts is set in the same way as the time chart and is restricted to numeric values.



Category Chart data properties.

## Totals Chart

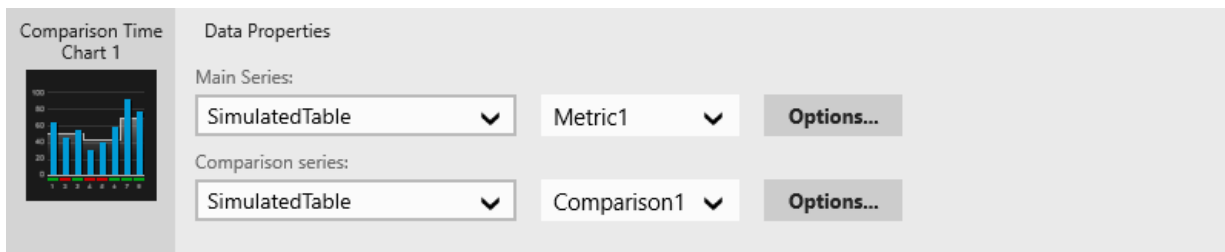
The totals chart accomplishes two unique things: firstly, it does not present multiple series but rather only the sum, or total, of the defined main series. Secondly, it has the option to use data table columns as a grouping when its data structure property is set to ByColumns. This can be quite useful when dealing with flattened data. In the latter mode, only the main series property is available as the category column is automatically determined by the number of fields selected for the main series property.



Totals Chart data properties.

## Comparison Chart

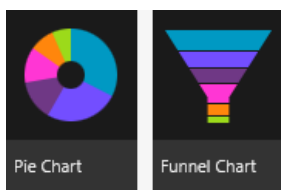
Each of the three chart types are available as comparison charts. This allows the user to specify not only a main series, but also a second comparison series which can be displayed in different ways by adjusting the series visualization to one of the following: BarVsThinBar, LineVsBar, and BarVsStepArea. These visualization types describe how the comparison series should be displayed in relation to the bars of the main series.



Comparison Chart data properties.

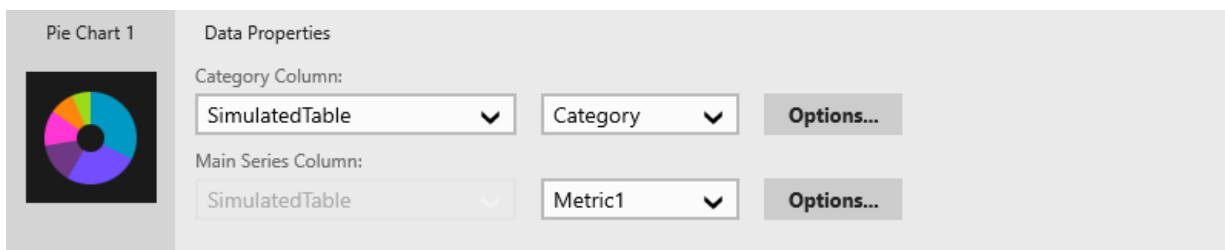
Comparison charts also introduce a unique visual property: reuse colors on comparison series. If set to true, the color palette used for the series will be reset between drawing the main and comparison Series, in order to coordinate between comparison series and similarly-sorted main series. If set to false, the color palette will continue its normal rotation when drawing the main series after the comparison series, preventing potentially misleading color coordination between the two sets of series.

## Pie and Funnel Charts



Pie and Funnel Chart visualizations.

Pie and funnel charts are among the simplest of visualizations. The data structure, found on the visual properties panel of the layout view, can either be set to ByRows or ByColumns. When set to ByColumns simply select the columns to aggregate and display on the chart in the data properties of the data view. The field names will be used to label each area of the resulting chart. Conversely, when using the ByRows data structure, a field must be specified for grouping and labels, the category column, as well as a numeric field for the main metric or the main series column of the chart.



Pie and Funnel Chart visualizations.

## TreeMap




TreeMap visualization.

The treemap element displays metrics by applying their values to the area and color of tiles within a rectangular grid. In

the data properties panel of the data view, a designer can select a numeric field to be represented by the tile's size and a numeric field to be represented by the tile's color. In addition to these properties, the designer can also set which fields should be displayed both in the popup of each tile and, when space permits, on the tile itself.

TreeMap 1



Data Properties

Size Represents:

SimulatedTreeMapTable

Metric1

Options...

Color Represents:

SimulatedTreeMapTable

Comparison1

Options...

Popup Labels:

SimulatedTreeMapTable

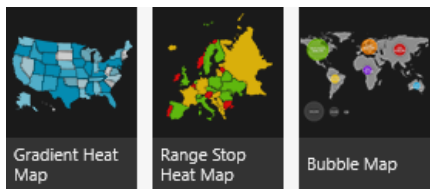
Metric1;...

Options...

TreeMap data properties.

A hierarchical treemap can be display by selecting the two-level treemap option in visual properties panel of the layout view. This will add the “group by” property to the data properties that should be set to a field by which the other metrics will be grouped and aggregated. Enabling two-level treemap works very well with show item headers enabled.

## Maps




Map visualizations.

Maps are a fantastic way to visualize geographical data, and Datzzen provides three types of map visualization along with 10 maps. These three elements require a few key data properties be set in order to properly display real data.

The first is the keys property which will connect the data to specific map regions. Ensure that you have the correct map selected – USA, for instance, if the data is indexed by USA states; or WorldCountries if the data is indexed by country name. The selected map may be changed in the visual properties panel in the layout view. Once the correct map has been selected the keys property in the data properties panel of the data view should be set to the data table and field which contains the region names, state or country for example.

Gradient Heat Map 1



Data Properties

Keys:

SimulatedUnitedStatesTab

MapKey

Options...

Values:

SimulatedUnitedStatesTab

Metric

Options...

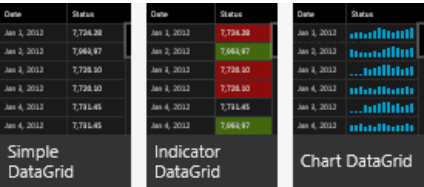
Map data properties.

The second property is the values property, which should be set to a numeric field in the same table as the selected keys field. These values will be represented in different ways depending on which map element is being used. The gradient map will use these values to color each region with a varying shades based on the range of values whereas the bubble map will base the size of a bubble visualization over each region on the value property. The bubble map also allows the dashboard designer to specify whether each region's bubble visualization should be a different color, or whether all should be the same color via the Use Different Colors toggle in the visual properties panel in the Layout

View.

The range stop heat map element allows the user to visualize a value's relation to a target. Because of this, the targets property must be set on the data properties panel of the data view, in addition to both values and keys. The targets property should be set to a numeric field in the same data table as both the values and keys. The range stop heat map also supports the value direction property, which may be set to either HigherValuesAreBetter or LowerValuesAreBetter. This determines whether positive values will be colored green, indicating a desirable change for the better, or red, indicating an undesirable change for the worse. The color of the regions will be determined by the sign of the delta between the value and the target as well as the value direction property setting.

DataGrids



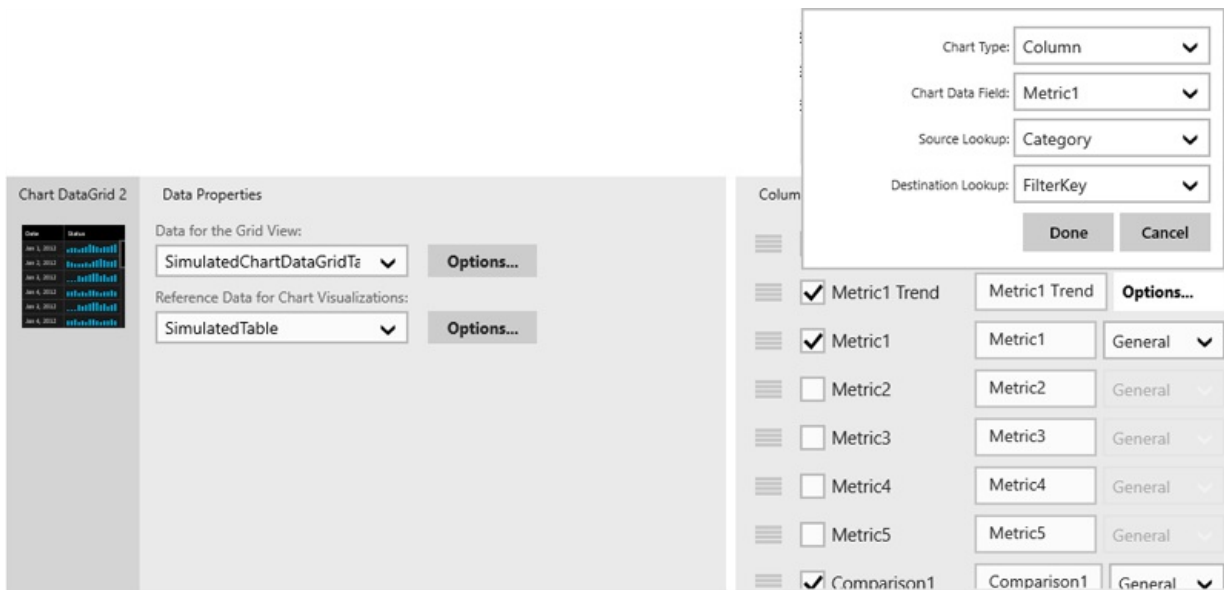
DataGrid visualizations.

In cases when the best visualization is the data itself, Datazen includes three datagrid elements to accomplish this.

The most basic datagrid, the simple datagrid, is capable of displaying multiple columns of data with custom formatting and headers. The simple datagrid can be connected to real data by selecting a data table in the data properties panel of the data view. Once a data table has been selected, the fields to be included can be selected by checking them off on the right hand side of the data properties panel. In addition to including column header text can be entered in the textboxes adjacent to the field checkboxes, and, in the case of date/time fields and numeric, formatting options can be specified.

The remaining two datagrid elements, the indicator datagrid and chart datagrid, allow the designer to integrate gauge and chart columns, respectively. This can be accomplished by selecting the “add gauge (or chart) column” in the columns panel of the data view.

Once a gauge column has been added to a datagrid some properties must be set in the options panel accessed by clicking the options button beside the gauge column in question. First, select a gauge type from the first drop-down list in the options panel, then continue to configure the gauge as normal. The datagrid will automatically feed the gauge only the data specific to that row of the datagrid.



DataGrid data properties.

Adding chart columns to datagrids requires additional considerations as a separate data table is needed to provide chart data for each row. This can be set in the data properties panel under “reference data for chart visualizations”. This second data table must share a field which may be joined in order to link each row to associated chart data.

After adding a chart column and selecting a chart type from the chart column options panel, each chart column must be assigned a data field, a source lookup, and a destination lookup. These three properties determine how the datagrid should provide data to each chart in the column.

Firstly, the source lookup property is set to a field in data table from which the datagrid itself is getting data. This field can be considered a per-row-filter which will be applied to the chart reference data table in order to provide data to the embedded chart for each row. After selecting a source lookup field from the datagrid’s source data table, select a destination lookup. This is the field in the “reference data for chart visualizations” data table. The data for the chart in each row will be joined on those two fields.

The last property that needs to be set is the chart data field. This determines which metric in the “reference data for chart visualizations” data table will be used as the y-axis value or series in the chart in each row.

# Publishing Dashboards

## Publishing Steps

To publish a dashboard to a server, open the desired dashboard in the designer, swipe up from the bottom of the screen (or right-click to activate the bottom menu) and click the publish button.

If there are no server connections configured, one must first be configured. Enter the appropriate connection credentials and click Connect.

If the dashboard has been edited since it was last saved, the first step in publishing a dashboard is to save it to the local folder. At this point, the title, filename, local group, culture, and fiscal start date can be modified. Make any desired changes and click Save.

Publish to Server, Step 1

Before being published to the server, your dashboard must be saved locally.  
Save dashboard as **New-Dashboard.datazen** ?

Dashboard Title:

New Dashboard

File Name:

New-Dashboard.datazen

Dashboard Group:

My Dashboards

Create New Group...

Dashboard Culture:

English (USA)

Affects default currency, date & number formatting, and static labels.

Fiscal Year Start:

January

1

Affects time navigators and time charts.

Save

Cancel

Publishing a Dashboard to a Server Step 1.

The next step in publishing a dashboard is to configure the server settings. Server configuration options are the server to connect to, dashboard title on the server, hub, and server group. Make any desired changes and click Publish.

Publish to Server, Step 2

Datazen Server:

Dashboard Title:

New Dashboard

Dashboard Hub:

Server Group:

Operations Dashboards

Create Server Group...

Publish

Cancel

Publishing a Dashboard to a Server Step 2.

At this point the dashboard has been published and will be accessible from any dashboard viewer client.

## **Saving a Dashboard Locally**

When publishing a dashboard it must first be saved locally as described above in Step 1. If you wish to save a dashboard locally without publishing this can be done at any point by swiping up or right clicking in the dashboard designer and selecting Save Locally.

Dashboards saved locally in your working directory will automatically appear on the main screen of Datazen. They may also be opened through the Windows file explorer by double-clicking on them.



## Advanced Topics

This section contains the following documents:

- [Parameterized Queries & Load on Demand](#)
- [Drill-throughs to Other Dashboards or Custom URLs](#)
- [Hierarchical Selections](#)
- [Cascading Selection Lists](#)
- [Working with Custom Maps](#)

# Parameterized Data Views and Load on Demand

## Creating parameterized data views

Start by creating a new data view inside a *Data Source* in the Control Panel. Currently, [parameterized views are supported only for SQL and MDX data types](#).

**Note:** You must be logged in as *admin* or the hub owner to configure data sources in a hub.

Check the *Real Time* checkbox and then click on the *Define Parameters* link beside it. Since parameterized queries are executed every time the view changes in a client application, they **must** be real time and cannot be [cached](#) in the Core Service Repository.

We can configure all the parameters needed to pass from the dashboard into our query in order to optimize it. The example below uses a parameterized query to return only results within a selected time range and only for a selected product.

```
SELECT * FROM [dbo].[ProductDataTable] WHERE [Month] BETWEEN
    CAST('{{ @StarTime }}' AS DATETIME)
    AND CAST('{{ @EndTime }}' AS DATETIME)
    AND [Product] LIKE CASE '{{ @Product }}'
    WHEN '' THEN '%' ELSE '{{ @Product }}'
END
```

Note that for the *Product* parameter we use the % symbol as a default value. This is a T-SQL wildcard that will match any string, meaning that we want all values returned when nothing is specified in the parameter.

Click on *Done* to create our query.

The SQL query can now make use of parameters by enclosing them in a set of double braces such as {{ @ParamName }}. Note that the *Product* WHERE clause has a special case for the empty string. The reason for this is that product selection will come from a Selection List control in a dashboard that has the *All* option enabled. When the user selects *All* the selection list will pass the empty string as a selection which we handle in our query to select all products.

Click on *Next* to finish creating our query.

## Creating dashboards with load on demand data

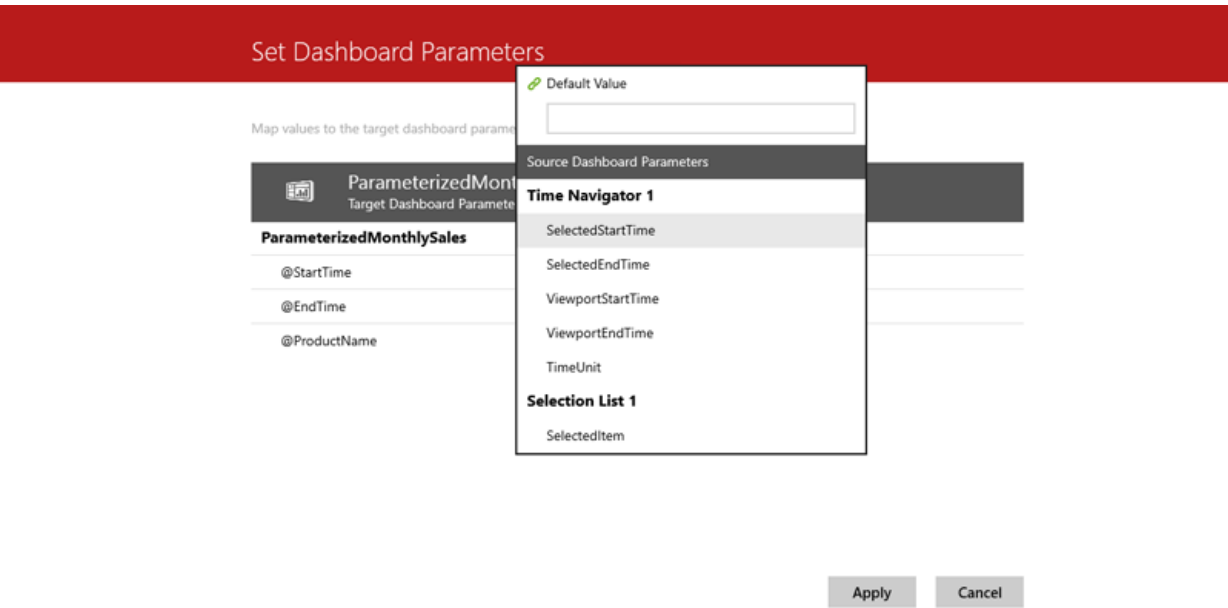
Creating dashboards that load data on demand is a two-step process:

1. Create the appropriate data views that allow parameters from the dashboards to be passed into the queries.
2. Connect the parameters to the appropriate filters in the dashboards in order for the view to be properly used by the dashboard.

## Using parameterized data views in a dashboard

In the *Data View* section of the Dashboard Designer parameterized data views can be imported from the server. Once imported a green *P* on the right of the data table's tab denotes that parameters are available to be configured for it. Clicking on the gear icon beside the green *P* and selecting the *Param* option will display the parameter configuration

screen.



Connecting data view parameters with navigators.

The parameter configuration screen allows for the binding of the data view's query parameters to parameters available from elements currently on a dashboard, such as Selection Lists and Time Navigators.

Once the parameter setup has been completed, each time one of the parameters changes through end user interaction with a dashboard, any data table with bound parameters will be refreshed, along with all the controls that use its data. With this approach, configuring filtering on the client is not needed since the filtering is done on the server through the parameters, and only the filtered results are returned.

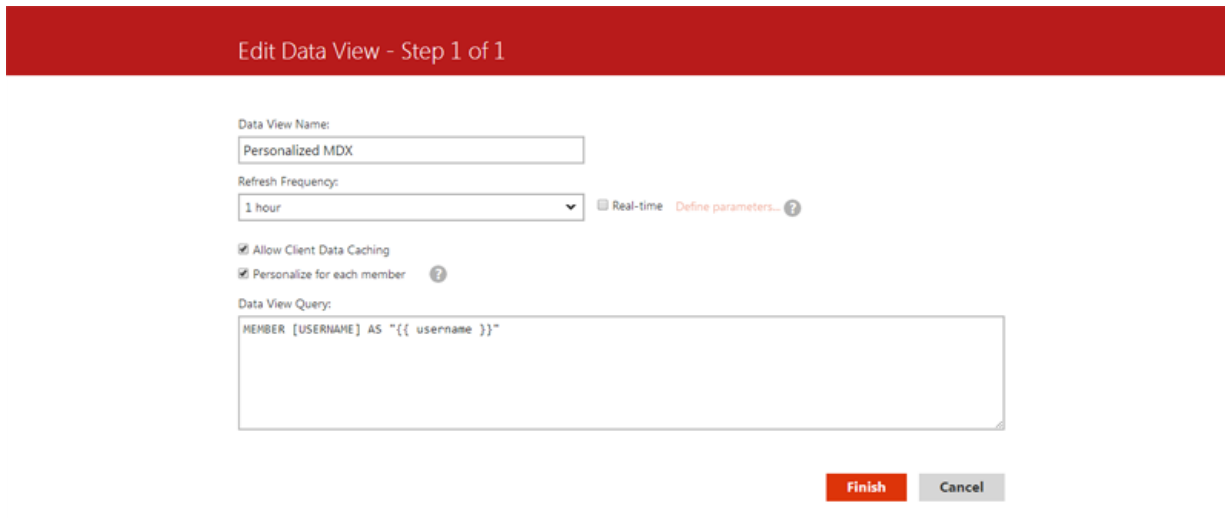
**Using parameterized data views with load on demand data**

Parameterized data views can be used to load data *on demand* by allowing the user to select filter parameters through the use of Selection Lists or Scorecard Grid, as well as allowing the user to select a time range by using a Time Navigator. Queries structured with *start* and *end* datetime parameters support load on demand data for any dashboards with a Time Navigator bound to those start and end datetime parameters.

# Configuring Personalized Data Views

All data views (with the exception of Excel documents), can be personalized based on the username of the currently loggen in user. The username is passed to the server using various methods depending on what is appropriate for the view type. Datazen supports personalized data views for both real time and cached views.

**Important:** When caching personalized data views the related query is executed and cached for **every user** with [access](#) to the data view's data connection.



Edit Data View - Step 1 of 1

Data View Name:  
Personalized MDX

Refresh Frequency:  
1 hour

☐ Real-time Define parameters... ?

☒ Allow Client Data Caching

☒ Personalize for each member ?

Data View Query:  
MEMBER [USERNAME] AS "{{ username }}"

Finish Cancel

An example of a personalized MDX query.

The username of the user which is accessing a data view is provided as a token, {{ username }}, to be included in the relevant query body.

## Personalized data view type

### XML

XML data views are queried through url parameters and therefore the current user will, in most cases, be passed as a querystring parameter:

```
http://mysoapserver.com/mysoapservice.aspx?user={{ username }}
```

### OData

OData queries can be personalized by including the username token in the query url, illustrated below in a filter parameter:

```
http://services.odata.org/Northwind/Northwind.svc/Customers?$filter=User%20eq%20'{{ username }}'
```

## SQL Server, SQL Azure, ODBC and OLEDB

In SQL-based personalized queries the username token is included directly in the text of the SQL query and can be used to provide a string comparison or an object key such as a table or column:

```
SELECT * FROM Table1 WHERE Username = "{{ username }}"
```

## SSAS

MDX queries can include the username token anywhere in the query, illustrated below by selecting a MEMBER by username:

```
MEMBER [USERNAME] AS "{{ username }}"
```

The username token can also be used, for example, in a SET statement.

SSAS queries can also be personalized through the *EffectiveUser* method utilizing row-level security. This method can be used in conjunction with the MDX method outlined above. For more information see [Implementing Row-Level Security for SSAS Data Views](#).

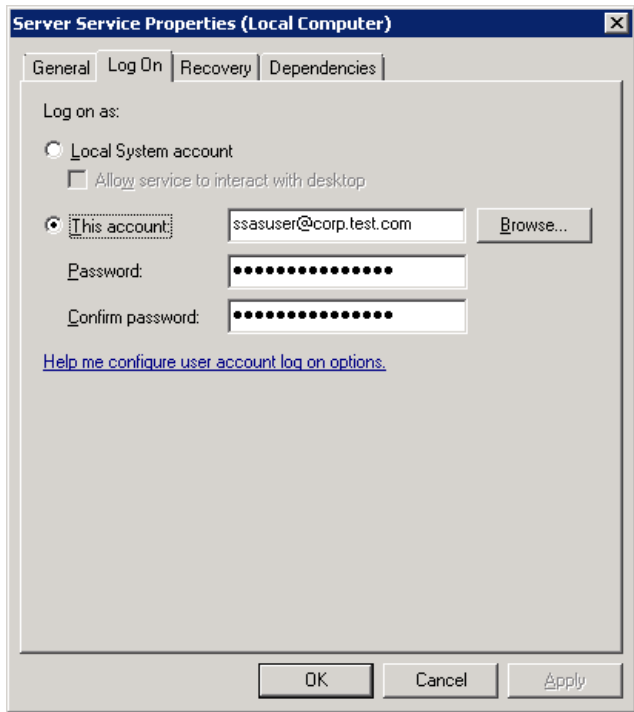
# Implementing Row-Level Security for SSAS Data Views

Datazen Enterprise Server supports row-level security for SQL Server Analysis Services (SSAS) data views in a Windows domain environment via the *EffectiveUserName* connection string property. For general information about Datazen Enterprise Server data personalization, see [Configuring Personalized Data Views](#).

## Configuring Core Service and Data Acquisition Service credentials

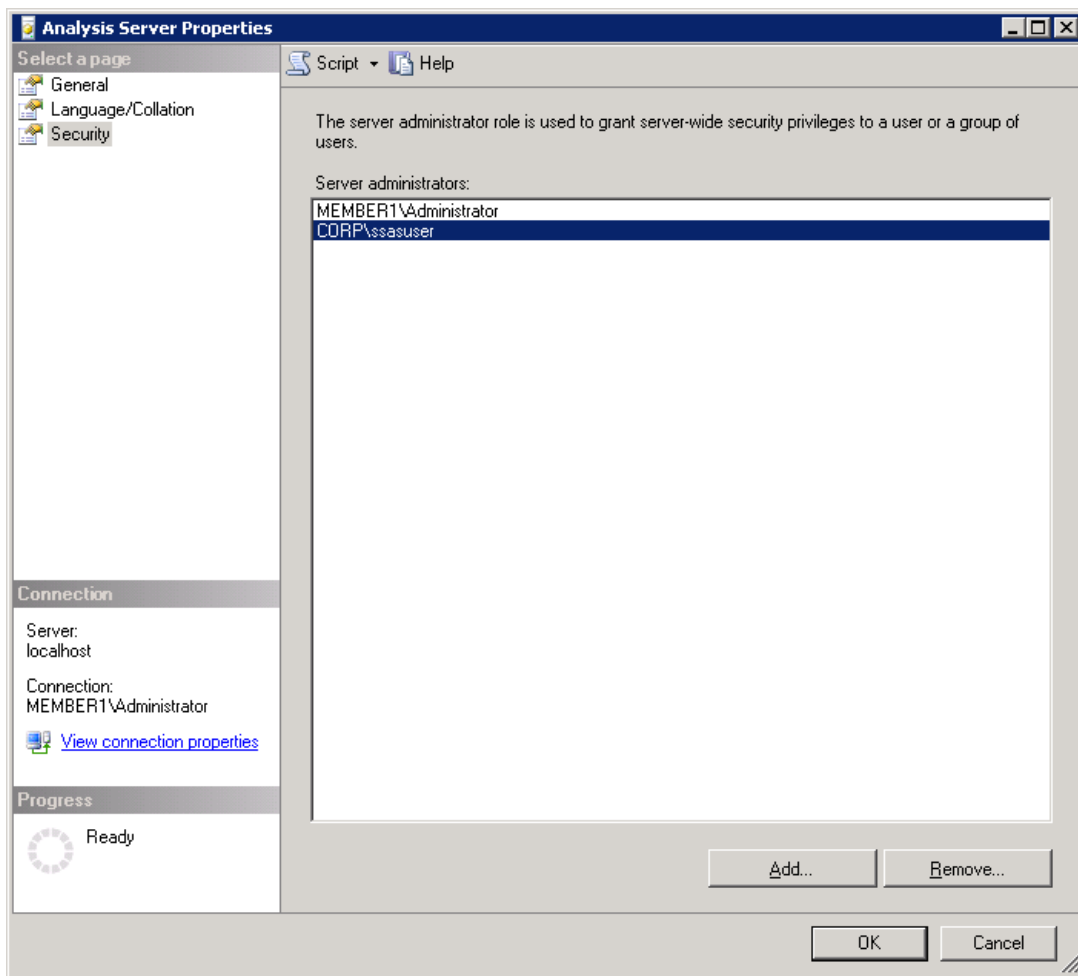
In order to utilize row-level security for SSAS, the Core Service and Data Acquisition Service must be configured to run under a domain account that has administrative privileges on the target SSAS instance.

If these credentials were *not* configured during [server installation](#), they can be configured through the Windows Services manager:

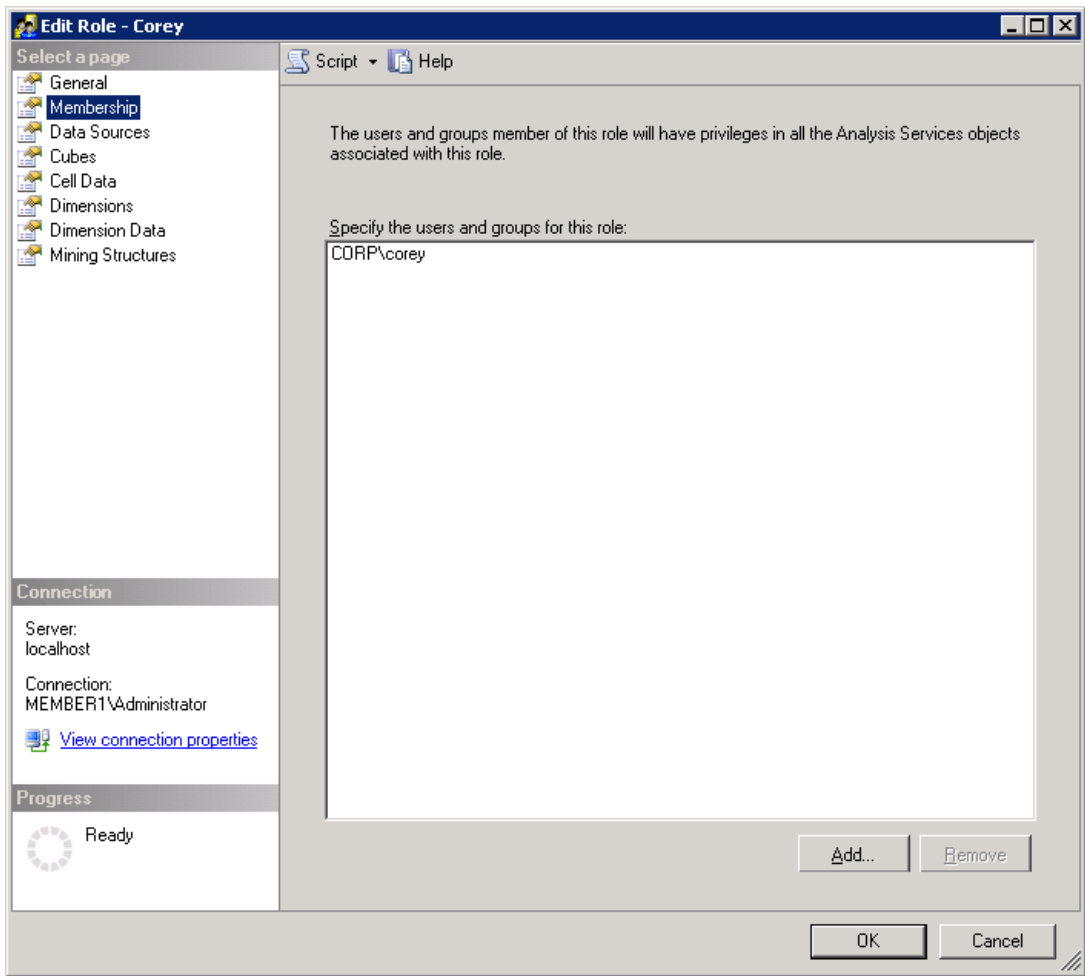


## Configuring SSAS security

The service credentials for Core Service and Data Acquisition Service can be added as a SSAS administrator with the SSAS Management Studio.



Every user that will have access to a personalized data connection must have access to the SSAS instance, either as an administrator or defined in a role.



Roles can then be used to secure SSAS data appropriately. Securing cube data with roles is outside the scope of this document.

### Creating personalized data connections

In the Datazen Server Control Panel, under *Data Sources*, create a new Analysis Services connection, check *Personalize with Effective User*, and provide the connection string.




## Edit Data Connection - Step 1 of 2

Data Connection Name:

SSAS (Personalized)

Personalization:

☒ Personalize with Effective User 

Properties:

Provider: MSOLAP

Data Source: .....

Catalog: .....

UserName:

Password:

Add a property...

Test Connection

Next

Cancel

When Datazen connects to Analysis Services, it will automatically append *EffectiveUserName={{ username }}* to the connection string. When any data view that belongs to this data connection is executed, it will run in the context of the specified user account.

**Note:** If you require more granularity over this setting in your connection string, you can manually set the *EffectiveUserName* property:

```
Provider=MSOLAP; Data Source=10.0.0.86; Initial Catalog=Adventure Works DW 2008R2 SE;EffectiveUserName=MYDOMAIN\{{ username }}
```

### Creating personalized data views

For further personalization, select *Personalize for each member* on the data view itself. In the query, any instance of the *{{ username }}* token will be substituted with the specific user's username.

## Edit Data View - Step 1 of 2

Data View Name:

Internet Product Sales (Personalized)

Refresh Frequency:

1 hour

☐ Real-time [Define parameters...](#) ?

☒ Allow Client Data Caching

☐ Personalize for each member ?

Data View Query:

```
WITH MEMBER [Product].[Category].[Date] AS [Date].[Calendar].CURRENTMEMBER.NAME
SELECT
NON EMPTY { [Product].[Category].[Date], [Product].[Category].[Category] } ON 0,
NONEMPTY({ [Date].[Calendar].[Date] }, { [Measures].[Internet Sales Amount] }) ON 1
FROM [Adventure Works]
WHERE { [Measures].[Internet Sales Amount] } * { STRTOMEMBER("[DimUsers].[Users].&[{{ username }}]") }
```

Next

Cancel

Implementing a user dimension in your cube is outside the scope of this document.

# Drill-throughs to Other Dashboards or Custom URLs

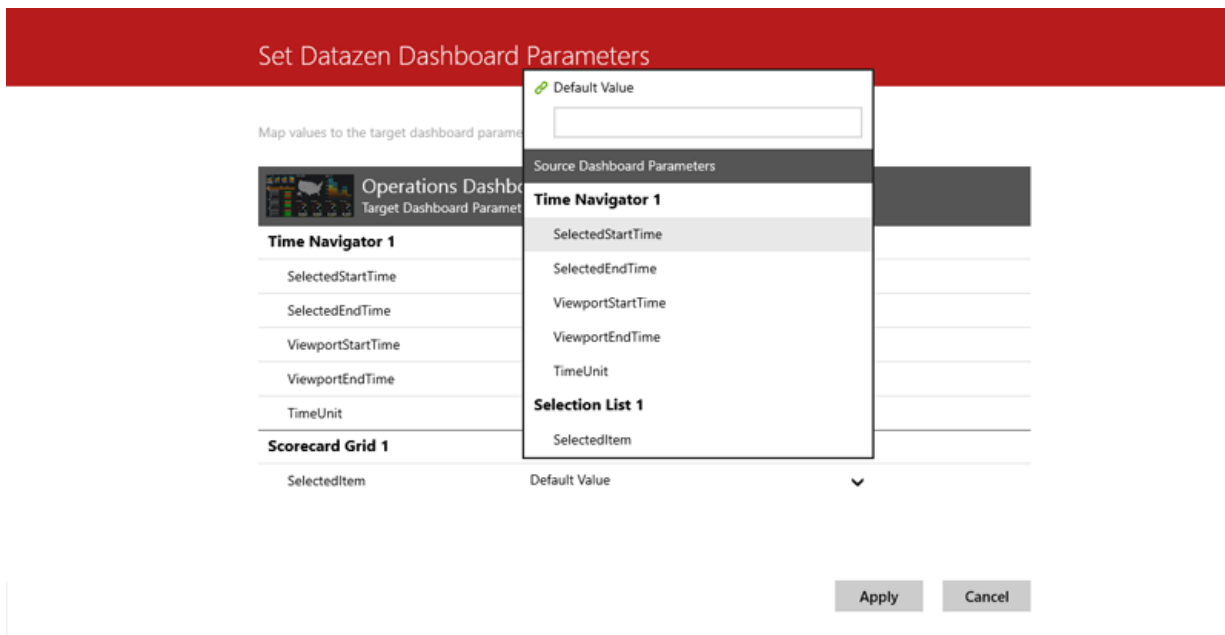
Any Gauge, Chart or Datagrid can be configured to initiate a drill-through action to another dashboard or custom URL. Depending on the source dashboard one or more parameters can be passed to the target dashboard or integrated into a custom URL.

At runtime a touch or click on an element configured with a drill-through target will send the user to the that target, either a dashboard or URL.

## Configuring a drill-through to another dashboard

To configure a drill-through to another dashboard, in the dashboard designer first select an element that support drill-through targets on the design surface. Open the Drill-through Target drop down in Visual Properties pane and select Dashboard, this will open the drill-through configuration dialog.

Select the desired target dashboard from the drill-through configuration dialog. Once a target dashboard has been selected it's available input parameters will be displayed, these parameters include properties which can be bound on navigator controls as well as any parameters configured on data tables of the target dashboard.



Drill-through to another dashboard properties.

Properties of matching data types can then be connected to available output properties on the source dashboard by using the dropdown to the right of each property. Defaults for each output may also be configured here in the event a user has not interacted with the dashboard before initiating a drill-through action.

When publishing dashboards drill-through target dashboards should be published before drill-throughs are configured on source dashboards.

## Configuring a drill-through to a custom URL

To configure a drill-through to a custom URL, in the dashboard designer first select an element that support drill-through targets on the design surface. Open the Drill-through Target drop down in Visual Properties pane and select CustomURL, this will open the drill-through configuration dialog.

In the Custom URL Parameter dialog a URL can be combined with available parameters by pasting, or typing the URL as well as touching or clicking on available parameter tokens listed to the right of the URL textbox. Parameters can also be typed or pasted manually into the URL textbox. A preview of the Custom URL combined with sample resolved parameters (if included) will be displayed in the panel below the input textbox.

Set Custom URL Parameters

Enter your expression. Parameter tokens will be replaced with parameter values.

www.server.com/reports/{{ SelectionList01.SelectedItem }}/{{ SimpleTimeNavigator01.SelectedStartTime }}

Available parameters:

{{ SelectionList01.SelectedItem }}

{{ SimpleTimeNavigator01.SelectedEndTime }}

{{ SimpleTimeNavigator01.SelectedStartTime }}

{{ SimpleTimeNavigator01.TimeUnit }}

{{ SimpleTimeNavigator01.ViewportEndTime }}

Here is a sample of how the effective target URLs will look. Parameter values seen here are for illustrative purposes only.

<http://www.server.com/reports/Key1/2001-01-01>

Apply

Cancel

Drill-through to a custom URL properties.

While previewing dashboards in the dashboard designer a dialog will be shown when drill-throughs are triggered. Drill-throughs can only be initiated when a dashboard is saved or published and viewed, not from within the dashboard designer or preview.

# Hierarchical Selections

## Preparing Hierarchical Data

The best practice for implementing hierarchical selections is to prepare data in a normalized fashion, specifically in the case of parent and child data relations.

Queries and tables should be structured to return data rows with a key, a parent key (or null) and a label. This data can then be used to create a hierarchical user interface with a hierarchically structured Selection List.

## Implementing a Hierarchical Selection List

Once data is prepared a hierarchical selection list may be implemented in a dashboard by dragging a Selection List onto the dashboard design surface and selecting Tree from the Data Structure dropdown in the Visual Properties panel below the design surface.

When a Selection List's Data Structure has been set to Tree the visual display of the element on the design surface will change to reflect a tree-like hierarchical structure. If the Selection List occupies only one cell vertically and thus displays as a dropdown the tree structure will not be visible at design time.

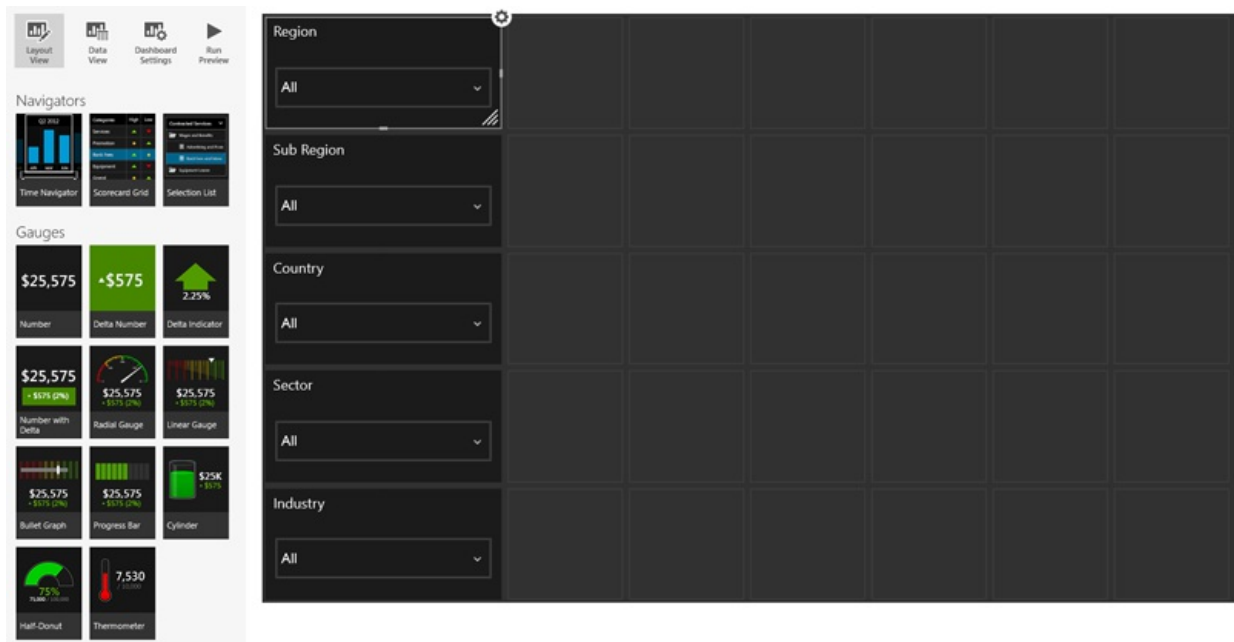
On the Data View of the dashboard designer a Selection List has properties for selecting columns to provide Key and Label. When set to use a Tree Data Structure an additional property is available for specifying a data table fields to provide Parent Keys. Once Keys, Parent Keys and Labels data properties have been set to appropriate data table fields the Selection List is ready to display a hierarchical structure.

While flattened or denormalized data may also be used to implement a hierarchical selection list, the preferred method is to use normalized data as outlined above.

# Cascading Selection Lists

When dealing with hierarchical data sets one of the options to allow user interaction is two or more cascading selection lists where each selection list is populated with one level of the hierarchy.

In the dashboard designer add a Selection List for each user selectable level in the hierarchy to the design surface. Selection lists occupying only one vertical cell on the design surface will be displayed as dropdowns.



Cascading Selection Lists arranged on the Design Surface.

In the Data View each Selection List instance's Key property, which can be found in the Data Properties panel in the lower portion of the screen, should be set to the Key or Id of the desired associated hierarchy level. The Label property can be set to the Key, or a related user friendly string field.

Each Selection List instance should also be set to be filtered by its parents in the hierarchy. This can be done by selecting the Options button to the right of the Key property for each Selection List and ensuring that each and every ancestor, preceding (or parent) Selection Lists is checked and therefore will filter the current Selection List. This must be done for each Selection List in the hierarchy.

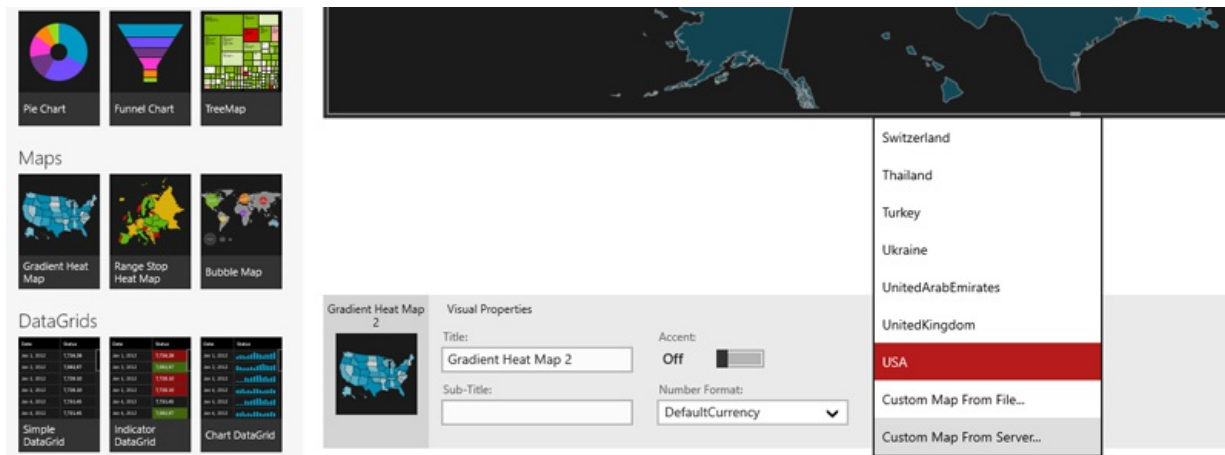
Setting filters for each Selection List should be done after all lists are in place and from the highest parent in the hierarchy to the lowest child. Cascading selection lists should be filtered by their parents and not by any child Selection Lists. Keeping this in mind, the top most Selection List should not be filtered by any others in the hierarchy. In this way a change to any Selection List in the hierarchy will propagate the filter throughout its descendants.

# Working with Custom Maps

Connecting data to Custom Maps requires that the data be prepared in advance so that at least one field matches the region keys which are used in the Custom Map. In order to use a Custom Map it must first be uploaded to the server by the Hub owner. Please refer to [Implementing Custom Maps](#) and [Managing Custom Maps](#) for information on creating and installing custom maps on a Datazen Enterprise Server.

## Loading a Custom Map

Once a Map gallery element has been placed on the dashboard design surface it can be connected to a Custom Map by selecting either Custom Map From File or Custom Map from Server. Selecting one of these options will allow you to browse either the server or your local file system for a Custom map.



Loading a custom map from a server or local file.

When loading a custom map from your local file system you must select *two* files: a file with the extension .SHP (shape) and a matching file with the extension .DBF (data). These two files are required for connecting data to a custom map. When loading a custom map from a server, however, each map will be presented as a *single* option. Selection of individual .SHP and .DBF files is not required from server sources.

## Connecting Data to a Custom Map

Three data properties are required for configuring Map instances, these are accessible from the Data View page of the dashboard designer.

In order to connect data to a custom Map the Keys property of the Map instance must be set to a data table field whose content matches the keys contained in the Map's data file. Map data keys in most cases will be region names.

Once Map keys are connected to data a Value field can be set, and in the case of the Range Stop Heat Map a Comparison field can be set as well.

Gradient Heat Maps display the Value properties as shades of a single color.

Bubble Map Value properties determine the radius of a bubble visualization displayed over the associated region.

Comparison fields for Range Stop Heat Maps will be used to determine the delta between the Comparison and the Value. The resulting delta will determine the color within a range which will fill the associated region of the map.

## Extending Datazen

This section contains the following documents:

- [Implementing Custom Branding](#)
- [Implementing Custom Maps](#)



# Implementing Custom Branding

A *brand package* can be created to modify the client application user interface (and optionally create a custom dashboard style) to match an organization's branding and aesthetics. To accomplish this a number of assets, images and configuration files are required.

Each brand package must contain the images for all supported clients:

- Windows 8,
- Android
- iOS
- Windows Phone
- HTML (web viewer)

Image files are named using a logical naming convention: *[target OS]-[file description]-[image dimensions].png*

Additionally, text-based configuration files must be created to specify:

- the colors of various user interface elements (buttons, borders, text fields, etc.)
- optionally, a custom dashboard style

Once all the images have been prepared and all configuration files edited, the assets will need to be assembled in a ZIP file and uploaded via the Control Panel web application.

## Image assets

Each supported client uses a variety of specifically-sized and named image assets. Some of these files are shared between clients, most of them exist solely for use in a discrete client.

As mentioned above, the majority of the images are logos of various sizes, while others are solid-color icons. These icons can be easily modified to fit any custom brand by changing their color.

**Important:** Unless otherwise specified, all graphic files are assumed to be 32-bit PNG images.

## Common

The following assets are shared between two or more different clients:

Filename	Dimensions	Description
Activity-42x42.png	42x42	Standard resolution activity button icon
Activity-42x42@2x.png	84x84 (42x42 * 200%)	High resolution activity button icon
BlankAvatarImage-80x80.png	80x80	Default user picture
Hub-Connected-32x32.png	32x32	Hub connection indicator

Hub-Disconnected–32x32.png	32x32	Hub “not connected” indicator
Server-Connected–64x64.png	64x64	Server connection indicator
Server-Default–64x64.png	64x64	Default server icon
Server-Disconnected–64x64.png	64x64	Server “not connected” indicator

## Android

The following unique files are required only by the Android client. Several different screen resolutions and configurations are supported.

Filename	Dimensions	Description
Android-AddNewServerIcon–30x30.png	30x30	Add New Server button icon
Android-Background–1200x1900.png	1200x1900	High resolution background image
Android-Background–768x1280.png	768x1280	Lower resolution background image
Android-Logo–320x64.png	320x64	Logo

## iOS

The following unique files are required by only the iOS clients. The application is a universal binary that runs on both iPhone and iPad. High-resolution (“Retina”) graphics are identified by the filename suffix @2x.

Filename	Dimensions	Description
iOS7-AddNewServerIcon–42x42.png	42x42	Standard resolution Add Server button icon
iOS7-AddNewServerIcon–42x42@2x.png	84x84	Retina resolution Add Server button icon
iOS7-Background–320x568.png	320x568	Standard resolution iPhone background (portrait)
iOS7-Background–320x568@2x.png	640x1136	Retina resolution iPhone background (portrait)
iOS7-Background–568x320.png	568x320	Standard resolution iPhone background (landscape)
iOS7-Background–568x320@2x.png	1136x640	Retina resolution iPhone background (landscape)
iOS7-Background–768x1024.png	768x1024	Standard resolution iPad background (portrait)
iOS7-Background–768x1024@2x.png	1536x2048	Retina resolution iPad background (portrait)

iOS7-Background-1024x768.png	1024x768	Standard resolution iPad background (landscape)
iOS7-Background-1024x768@2x.png	2048x1536	Retina resolution iPad background (landscape)
iOS7-BackIcon-42x42.png	42x42	Standard resolution Back button icon
iOS7-BackIcon-42x42@2x.png	84x84	Retina resolution Back button icon
iOS7-Logo-225x45.png	225x45	Standard resolution logo
iOS7-Logo-225x45@2x.png	550x90	Retina resolution logo
iOS7-Menu-32x32.png	32x32	Standard resolution Menu button icon
iOS7-Menu-32x32@2x.png	64x64	Retina resolution Menu button icon
iOS7-Search-32x32.png	32x32	Standard resolution Search button icon
iOS7-Search-32x32@2x.png	64x64	Retina resolution Search button icon
iOS7-Settings-42x42.png	42x42	Standard resolution Settings button icon
iOS7-Settings-42x42@2x.png	84x84	Retina resolution Settings button icon

## Windows 8

The following unique files are required only by the Windows 8 client. Note that in addition to the standard naming convention, a resolution identifier (.100, .140, and .180) immediately precedes the file extension. These numbers represent the size of each image as a percentage of the standard resolution image and each asset should therefore be sized appropriately: 100%, 140%, and 180%. These images are then loaded automatically based on the client's display capabilities.

Filename	Dimensions	Description
Windows8-Background-1366x768.100.png	1366x768	Standard resolution background image
Windows8-Background-1366x768.140.png	1912x1075 (1366x768 * 140%)	Medium resolution background image
Windows8-Background-1366x768.180.png	2459x1382 (1366x768 * 180%)	High resolution background image
Windows8-Footer-220x22.100.png	220x22	Standard resolution "Powered by..." logo
Windows8-Footer-220x22.140.png	308x31 (220x22 * 140%)	Medium resolution "Powered by..." logo

Windows8-Footer–220x22.180.png	396x40 (220x22 * 180%)	High resolution “Powered by...” logo
Windows8-Logo–450x90.100.png	450x90	Standard resolution main logo
Windows8-Logo–450x90.140.png	630x126 (450x90 * 140%)	Medium resolution main logo
Windows8-Logo–450x90.180.png	810x162 (450x90 * 180%)	High resolution main logo
Windows8-SmallLogo–150x30.100.png	150x30	Medium resolution main logo
Windows8-SmallLogo–150x30.140.png	210x42 (150x30 * 140%)	Medium resolution smaller logo
Windows8-SmallLogo–150x30.180.png	270x54 (150x30 * 180%)	High resolution smaller logo
Windows8-Style–120x110.100.png	120x110	(Optional) Standard resolution style preview image
Windows8-Style–120x110.140.png	168x154 (120x110 * 140%)	(Optional) Medium resolution style preview image
Windows8-Style–120x110.180.png	216x198 (120x110 * 180%)	(Optional) High resolution style preview image

**Important** If no custom dashboard style is required, ensure that all Window8-Style files are omitted from the final brand package ZIP file. This includes the three PNGs as well as the Window8-Style.xaml config file.

## Windows Phone

The following unique files are required only by the Windows Phone clients. Note that while high-resolution image filenames are appended with the resolution identifier .150, this number does not reflect the image’s scale. And unlike Windows 8 assets, standard resolution image filenames do not use a .100 identifier.

Filename	Dimensions	Description
WP-Background–480x853.png	480x853	Standard resolution background image
WP-Background–480x853.150.png	1080x1920	Very high resolution background image
WP-Logo–280x40.png	280x40	Standard resolution logo
WP-Logo–280x40.150.png	560x80	High resolution logo

## HTML

Only three images are required for the two HTML pages: a standard resolution logo, a high resolution logo, and a spinner image. For the sake of ease, the two required logos can be renamed copies of the similar iOS assets, iOS7-Logo

-225x45.png and iOS7-Logo-225x45@2x.png. The spinner image, a 32-bit PNG, is animated using a CSS rotation transform and is displayed very briefly, so a high resolution version of the asset is unnecessary. It is recommended that the spinner image be kept as it is, but recolored as appropriate.

Filename	Dimensions	Description
Html-Logo-225x45.png	225x45	Standard resolution logo
Html-Logo-225x45@2x.png	550x90	High resolution logo
Html-Spinner-42x42.png	42x42	Loading spinner image, animated in-browser via CSS transform.

## Configuration files

A brand package requires two text-based configuration files. A third optional file *Windows8-Style.xaml* is needed if a custom dashboard style is desired.

Filename	Description
theme.json	A JSON file containing the name/value pairs for setting the colors of the client application's main user interface.
Html-Style.css	A CSS file for styling the browser-based login and dashboard list pages.
Windows8-Style.xaml	An optional XAML file for styling for creating custom dashboard styles and color palettes.

## Creating a custom dashboard style (Windows8-Style.xaml)

A custom brand package can be easily extended with the addition of a custom dashboard style (color palettes and UI styles that affect the appearance of dashboards).

In order to create a custom a style, the *Windows8-Style.xaml* configuration file will require editing. In addition three PNG image assets, mentioned in the above Windows 8 section, are required. These are the style thumbnails used in the dashboard style dropdown menu.

The XML object described within the XAML file is divided into three discrete parts:

- main properties
- color palette
- dashboard panel styles.

**Important** Properties not described in the list below should **not** be modified.

### Main properties

The main (root) properties of the style definition are further divided into three groups

- general style colors
- standard dashboard style colors
- accent dashboard style colors

## General style colors

Three hexadecimal properties handle general colors: *TileBrush*, *PositiveValueBrush*, and *NegativeValueBrush*. *TileBrush* controls the color of the dashboard tile, the small rectangle containing the title and thumbnail that appears in the dashboard list view. *PositiveValueBrush* and *NegativeValueBrush* control the colors for positive values and negative values throughout a dashboard. These are generally shades of green and red, respectively.

## Standard dashboard style colors

These properties handle the default appearance of dashboards using this style. *Background*, a hexadecimal value, is the dashboard's background color. *TitleBrush* and *SubTitleBrush*, both hex colors, are used for the main dashboard's title and subtitle. *PanelVariant* and *PanelBackground* handle the appearance of dashboard panels - the boxes that surround dashboard controls. A dashboard panel can have its own title and subtitle colors, as described further below. Finally, *SelectedForeground* and *SelectedBackground* handle the appearance of selected elements in Selection Lists (both List and Tree styles).

A string value, *PanelVariant*, tells the dashboard which base theme variant to use to draw the dashboard panels. Valid properties are *Standard* or *Accent*. A *Standard* panel is considered to have a light-on-dark appearance, while an *Accent* panel has a "dark-on-light" color appearance. Choose the value most appropriate for this particular style.

## Accent dashboard style colors

These properties handle the accent appearance of dashboards using this style. *AccentBackground*, a hexadecimal value, is the dashboard's background color. *AccentTitleBrush* and *AccentSubTitleBrush*, both hex colors, are used for the main dashboard's title and subtitle. *AccentPanelVariant* and *AccentPanelBackground*, handle the appearance of dashboard panels - the boxes that surround dashboard controls. A dashboard panel can have its own title and subtitle colors, as described further below. Finally, *AccentSelectedForeground* and *AccentSelectedBackground* handle the appearance of selected elements in Selection Lists (both List and Tree styles).

A string value, *AccentPanelVariant*, tells the dashboard which base theme variant to use to draw the dashboard panels. Valid properties are *Standard* or *Accent*. A *Standard* panel is considered to have a light-on-dark appearance, while an *Accent* panel has a "dark-on-light" color appearance. Choose the value most appropriate for this particular style.

## The color palette

At the heart of a dashboard style is the color palette definition. These are one or more comma-separated hexadecimal values representing the colors used by various controls (charts, gauges, maps) to visually identify data.

### PaletteName

A globally unique, valid string id to represent the color palette, eg. `PaletteName="MyCompanyGreen"` or `PaletteName="MyCompanyBlue"`.

## ChartingDataPoints

These collections of hexadecimal values represent the colors used by a chart when it contains a specific number of data point series. If, for example, the control is given four series, it will draw the data points using the colors from the *ChartingDataPoints4* collection.

It is recommended to create the largest collection, *ChartingDataPoints12*, first and then remove color for the lesser properties.

## GaugeStops and MappingStops

These collections of three colors handle the colors for their respective controls.

*GaugeStops* contains the colors for the three different gauge sections in the following order:

- Positive
- Caution
- Negative

These are generally green, yellow, and red respectively.

*MappingStops* contains the colors used by the map control to differentiate different data regions.

Using the same values found in *ChartingDataPoints3* is recommended.

## Dashboard panel styles

Due to the way the dashboard style parser is designed, four properties must be set to complete a dashboard style: two each for the default dashboard panel style and the accent dashboard panel style. These properties handle the color of the title and subtitle within an individual panel. These properties are set among the two template code blocks:

*PanelHeaderTemplate* and *AccentPanelHeaderTemplate*.

### Title

To set the color of the title of a dashboard panel, locate and change the *Foreground* property of a *TextBlock* with the property `Text="{Binding Title}"` found within the corresponding *PanelHeaderTemplate* block.

### SubTitle

To set the color of the subtitle of a dashboard panel, locate and change the *Foreground* property of a *TextBlock* with the

property Text="{Binding SubTitle}" found within the corresponding *PanelHeaderTemplate* block.

## **Style thumbnail images**

Once the XAML file has been edited, two images are required. These will act as thumbnails within the dashboard style dropdown menu. Standard resolution, medium resolution, and high-resolution images are required. Please refer to the image information described in the Windows 8 images list in this document.

## **Creating the final brand package**

Once all the images and configuration files have assembled, a single brand package file must be created. This is done by creating a ZIP archive containing the required files. In the ZIP archive, files must exist on the root level (the archive cannot contain sub-folders).

Once this file is created, it should be uploaded and assigned using the Control Panel web application. See [Managing Custom Branding](#) for more information.



# Implementing Custom Maps

Geographic maps in Datazen are defined in a format known as *ESRI shapefiles*.

Initially designed by a private company, this is now a widespread semi-open format used in a large portion of GIS applications. In accordance with this format, Datazen requires two files to be provided when defining a map:

- a .SHP file for shape geometries
- a .DBF file for meta data

The base files names must match (e.g. *canada.shp* and *canada.dbf*). The metadata must include the field *NAME* with the value of the corresponding shape's name (key), to be used when populating the map with data.

For information on how to make custom maps available for inclusion in dashboards, see [Managing Custom Maps](#).

## Information & Resources

More specific information can be found by visiting some of the following links:

### Technical information

- The official specification: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>
- The Wikipedia shapefile article: <http://en.wikipedia.org/wiki/Shapefile>

### Creating & editing map geometry

Creating and editing shapefiles is a complex process that is beyond the scope of this document. Here are some resources and applications to help you get started:

- ArcGIS: <http://www.arcgis.com/>
- MAPublisher plug-in for Adobe Illustrator: <http://www.avenza.com/mapublisher>
- QuantumGIS (free): <http://www.qgis.org/>
- Manco ShapeFile Editor: <http://www.mancosoftware.com/ShapeFileEditor>

### Existing shapefiles

Many existing shapefiles can be downloaded from the Web, from sites like these:

- Diva-GIS: <http://www.diva-gis.org/Data>
- OpenStreetMap: <http://openstreetmapdata.com/data>
- GeoCommons: <http://www.geocommons.com/>

# Managing Data Providers

Many [data providers](#) are available for use within Datazen. The set of *built-in* data providers can be extended by providing additional *custom* data providers. Also, specific data providers can be administratively disabled so that they will not be available to data connection authors.

## Implementing custom data providers

To create a custom data provider, a schema definition file representing the custom data provider should be created in the `\service\dataproverschemas` folder (or a sub-folder thereof) of the installation folder.

The schema definition file is an XML document in the following format:

```
<dataproverschema>
  <id>string</id>
  <enabled>true|*false*</enabled>
  <name>string</name>
  <type>sql|sqlazure|ssas|odbc|oledb</type>
  <properties>
    <property>
      <name>string</name>
      <aliases>
        <alias>string</alias>
        <alias>...</alias>
      </aliases>
      <value>string</value>
      <type>*string*|number|boolean</type>
      <readonly>true|*false*</readonly>
      <optional>true|*false*</optional>
      <masked>true|*false*</masked>
      <maskedForNonOwner>true|*false*</maskedForNonOwner>
      <allowPathTraversal>true|*false*</allowPathTraversal>
      <startsWith>string</startsWith>
      <pattern>regexp</pattern>
      <options>
        <option>string</option>
        <option>...</option>
      </options>
    </property>
    <property>
      ...
    </property>
  </properties>
</dataproverschema>
```

The unique id of the data provider. If more than one schema definition file has the same *id*, the last one loaded will be used (according to the alphanumerically sorted full path and file name of the schema definition file). It is recommended that the format of the *id* start with the underlying data provider driver type and optionally include the driver name and/or the intended audience (e.g. ODBC.SQLITE, or MSSQL.FINANCEDEPT). This element is required.

Location:

```
<dataproviderschema>
  <id>[string]</id>
```

## enabled

Determines whether the data provider is visible in the Control Panel. To disable a built-in data provider, use the *id* of the built-in data provider (as listed in the table below) and set *enabled* to *false*. This element is required.

Location:

```
<dataproviderschema>
  <enabled>[true|false]</enabled>
```

## name

The name of the data provider as it appears in the Control Panel. This element is required when *enabled* is true.

Location:

```
<dataproviderschema>
  <name>[string]</name>
```

## type

The underlying driver type of the data provider. This element is required when *enabled* is true.

Location:

```
<dataproviderschema>
  <type>[sql|sqlazure|ssas|odbc|oledb]</type>
```

## properties

A collection of *property* nodes that define the components of the connection string. This element is required when *enabled* is true.

Location:

```
<dataproviderschema>
  <properties>
    ....
```

## property name

The name of the property. Example: Data Source, or Server. This element is required.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <name>[string]</name>
```

## property aliases

A collection of *alias* nodes that define alternate names for the property. Example: Database.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <aliases>
        <alias>[string]</alias>
        ...
```

## property value

The default value of the property. When *readonly* is true and *optional* is false this value becomes mandatory in the data connection.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <value>[string]</value>
```

## property type

The data type of the property.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <type>[boolean|number|string]</type>
```

Default value: string

## property readonly

Marks this property as read-only.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <readonly>[true|false]</readonly>
```

Default value: false

## property optional

Optional properties can be added or removed when constructing a data connection.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <optional>[true|false]</optional>
```

Default value: false

## property masked

Whether the property will be masked in the Control Panel. Masked properties are protected as password-style inputs only when *type* is string. Unless *readonly* is set to true, new values may still be set.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <masked>[true|false]</masked>
```

Default value: false

## property maskedForNonOwner

Whether the property will be masked in the Control Panel for users that are not a hub owner. Masked properties are protected as password-style inputs only when *type* is string.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <maskedForNonOwner>[true|false]</maskedForNonOwner>
```

Default value: false

## property allowPathTraversal

When *type* is string, the property *value* is validated against potential path traversal attacks. When *allowPathTraversal* is false, any value with ..\, or ../, will be rejected. **It is strongly recommended that *allowPathTraversal* be set to false.**

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <allowPathTraversal>[true|false]</allowPathTraversal>
```

Default value: false

## property startsWith

A string that the property *value* must start with. Any value that does not start with the specified string will be rejected.

For example, if the property is a location to store log files, a file path starting with c:\logs\ can be enforced.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <startsWith>[string]</startsWith>
```

## property pattern

A regular expression that the property *value* must match. Any value that does not match the specified pattern will be rejected.

For example, if the property *value* must start with either c:\logs\ or \\logserver\logs\ and end with .log, *pattern* would be `^(c:\\logs\\|\\\\logserver\\logs\\)w+\\.log$`

Location:

```
<dataproviderschema>
  <properties>
```

```
<property>
  <pattern>[regular expression]</pattern>
```

## property options

A collection of *option* elements that define acceptable values for the property. If set, the property *value* must match one of the values in the collection, otherwise it will be rejected.

Location:

```
<dataproviderschema>
  <properties>
    <property>
      <options>
        <option>[string]</option>
        ...
```

Note: All other validation rules are still applied when *options* are specified.

## Built-in data providers

The following table describes the built-in data provider schemas:

ID	Type	Description
MSSQL	sql	Microsoft SQL Server
MSSQLAZURE	sqlazure	Microsoft Azure SQL Database
MSSAS	ssas	Microsoft SQL Server Analysis Services
ODBC.DSN	odbc	Generic ODBC DSN
ODBC.MYSQL	odbc	MySQL ODBC Driver
ODBC.ORACLE	odbc	Oracle ODBC Driver
ODBC.POSTGRES	odbc	PostgreSQL ODBC Driver
OLEDB.ORACLE.ORAOLEDB	oledb	Oracle Provider for OLE DB
OLEDB.POSTGRES.PGPNP	oledb	PGNP OLE DB Provider for PostgreSQL

## Overriding built-in data providers

It is possible to override a built-in data provider by creating a custom schema definition file with the same *id* as a built-in data provider. Overridden data providers should be used to include new properties, modify existing properties, or remove existing properties.

## Disabling built-in data providers

To disable a built-in data provider, create a custom schema definition file with the same *id* of the built-in data provider and set the *enabled* value to false.

For example, to disable the built-in Microsoft SQL Server Analysis Services data provider, create a custom schema definition file with the following XML:

```
<dataproviderschema>
  <id>MSSAS</id>
  <enabled>false</enabled>
</dataproviderschema>
```

## Locking down data providers

If the server administrator wishes to lock down the set of data providers so that data connection authors may only select a provider from a predefined set, disable each built-in data provider and create a schema definition file for each specific data connection scenario.

For example, to create a data provider for MS SQL Server to a specific database:

```
<dataproviderschema>
  <id>MSSQL.FINANCEDEPT</id>
  <enabled>true</enabled>
  <name>Accounting Database</name>
  <type>sql</type>
  <properties>
    <property>
      <name>Data Source</name>
      <value>finance.corp.domain.com</value>
      <masked>true</masked>
      <readonly>true</readonly>
    </property>
    <property>
      <name>Initial Catalog</name>
      <value>Accounting</value>
      <masked>true</masked>
      <readonly>true</readonly>
    </property>
    <property>
      <name>User ID</name>
      <value>uacct</value>
      <masked>true</masked>
      <readonly>true</readonly>
    </property>
    <property>
      <name>Password</name>
```



```
        <value>Pa$$w0rd</value>
        <masked>true</masked>
        <readonly>true</readonly>
    </property>
    <property>
        <name>Encrypt</name>
        <type>boolean</type>
        <value>true</value>
        <readonly>true</readonly>
    </property>
</properties>
</dataproviderschema>
```

Because every property is masked and read-only, a data connection author will not be able to see or edit the property values when creating a new connection with this provider.