



# DigitalEdge™

## Operations Guide

## Version 1.2.1

## July 2014



© Leidos. All rights reserved.

## **DISCLAIMER OF WARRANTY AND LIMITATION OF LIABILITY**

The Software accompanying this Documentation is provided with the Limited Warranty contained in the License Agreement for that Software. Leidos, its affiliates and suppliers, disclaim all warranties that the Software will perform as expected or desired on any machine or in any environment. Leidos, its affiliates and suppliers, further disclaim any warranties that this Documentation is complete, accurate, or error-free. Both the Software and the Documentation are subject to updates or changes at Leidos' sole discretion. LEIDOS, ITS LICENSORS AND SUPPLIERS MAKE NO OTHER WARRANTIES, WRITTEN OR ORAL, EXPRESS OR IMPLIED RELATING TO THE PRODUCTS, SOFTWARE, AND DOCUMENTATION. LEIDOS, ITS LICENSORS AND SUPPLIERS DISCLAIM ALL IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, USE, TITLE, AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. In no event shall Leidos, its affiliates or suppliers, be liable to the End User for any consequential, incidental, indirect, exemplary, punitive, or special damages (including lost profits, lost data, or cost of substitute goods or services) related to or arising out of the use of this Software and Documentation however caused and whether such damages are based in tort (including negligence), contract, or otherwise, and regardless of whether Leidos, its affiliates or suppliers, has been advised of the possibility of such damages in advance. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAWS, END USER ACKNOWLEDGES AND AGREES THAT LEIDOS AND ITS AFFILIATES AND SUPPLIERS IN NO EVENT SHALL BE RESPONSIBLE OR LIABLE TO THE END USER FOR ANY AMOUNTS IN EXCESS OF THE FEES PAID BY THE END USER TO LEIDOS. LEIDOS SHALL NOT BE RESPONSIBLE FOR ANY MATTER BEYOND ITS REASONABLE CONTROL.

## **LEIDOS PROPRIETARY INFORMATION**

This document contains Leidos Proprietary Information. It may be used by recipient only for the purpose for which it was transmitted and will be returned or destroyed upon request or when no longer needed by recipient. It may not be copied or communicated without the advance written consent of Leidos. This document contains trade secrets and commercial or financial information which are privileged and confidential and exempt from disclosure under the Freedom of Information Act, 5 U.S.C. § 552.

## **TRADEMARKS AND ACKNOWLEDGMENTS**

Private installations of DigitalEdge are powered by Eucalyptus®.

Public cloud installations of DigitalEdge are powered by Amazon Web Services™.

The following list includes all trademarks that are referenced throughout the DigitalEdge documentation suite.

Adobe, Flash, PDF, and Shockwave are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Amazon Web Services, AWS, Amazon Elastic Compute Cloud, Amazon EC2, EC2, Amazon Simple Storage Service, Amazon S3, Amazon VPC, Amazon DynamoDB, Amazon Route 53, the "Powered by Amazon Web Services" logo, are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

Apache, Archiva, Cassandra, Hadoop, Hive, HBase, Hue, Lucene, Maven, Apache Phoenix, Solr, Zoie, ActiveMQ are all trademarks of The Apache Software Foundation.

ArcSight is a registered trademark of ArcSight, Inc.

CAS is copyright 2007, JA-SIG, Inc.

CentOS is a trademark of the CentOS Project.

Cloudera is a registered trademark of Cloudera, Inc.

CloudShield is a registered trademark of CloudShield Technologies, Inc. in the U.S. and/or other countries.

CTools are open-source tools produced and managed by Webdetails Consulting Company in Portugal.

Drupal is a registered trademark of Dries Buytaert.

Elasticsearch is a trademark of Elasticsearch BV, registered in the U.S. and in other countries.

Eucalyptus and Walrus are registered trademarks of Eucalyptus Systems, Inc.

Firefox is a registered trademark of the Mozilla Foundation.

The Groovy programming language is sustained and led by SpringSource and the Groovy Community.

H2 is available under a modified version of the Mozilla Public License and under the unmodified Eclipse Public License.

Hybridfox is developed and maintained by CSS Corp R&D Labs.

JUnit is available under the terms of the Common Public License v 1.0.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft, Windows, and Word are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

MongoDB and Mongo are registered trademarks of 10gen, Inc.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Pentaho is a registered trademark of Pentaho, Inc.

PostgreSQL is a trademark of The PostgreSQL Global Development Group, in the US and other countries.

PuTTY is copyright 1997-2012 Simon Tatham.

Sonatype Nexus is a trademark of Sonatype, Inc.

Tableau Software and Tableau are registered trademarks of Tableau Software, Inc.

Twitter is a registered trademark of Twitter, Inc.

All other trademarks are the property of their respective owners.

## **CONTACT INFORMATION**

Leidos Franklin Center  
6841 Benjamin Franklin Drive  
Columbia, Maryland 21046

Email: [DigitalEdgeSupport@Leidos.com](mailto:DigitalEdgeSupport@Leidos.com)

DigitalEdge Technical Support: 443-367-7770

DigitalEdge Sales Support: 443-367-7800

To submit ideas or feedback: <https://www9.v1ideas.com/digitaledge/welcome>

# Contents

<b>Chapter 1: Introduction</b>	<b><a href="#">1</a></b>
Product documentation	<a href="#">1</a>
Typographical conventions	<a href="#">1</a>
<b>Chapter 2: DigitalEdge Architecture</b>	<b><a href="#">3</a></b>
Tenant Management System (TMS)	<a href="#">4</a>
About the gateway node	<a href="#">6</a>
About the master node	<a href="#">6</a>
<b>Chapter 3: Security</b>	<b><a href="#">7</a></b>
Certificates	<a href="#">7</a>
Securing data	<a href="#">7</a>
Security markings	<a href="#">7</a>
Security perimeters	<a href="#">8</a>
Management Console security	<a href="#">8</a>
Logs	<a href="#">8</a>
User auditing	<a href="#">8</a>
User management and administration	<a href="#">8</a>
User authentication	<a href="#">8</a>
<b>Chapter 4: Getting Data into the System</b>	<b><a href="#">11</a></b>
Getting data in with transports	<a href="#">11</a>
Transport examples	<a href="#">13</a>
Getting unstructured data in	<a href="#">15</a>
Getting data in with the Data Transfer Utility	<a href="#">17</a>
Generating metrics for CSV files	<a href="#">24</a>
<b>Chapter 5: Managing DigitalEdge from the Management Console</b>	<b><a href="#">29</a></b>
Logging in	<a href="#">30</a>
Starting a system for the first time	<a href="#">30</a>
Shutting down and restarting a system	<a href="#">31</a>

Stopping and restarting a process group .....	<a href="#">32</a>
Updating a system .....	<a href="#">33</a>
Deleting a system .....	<a href="#">34</a>
Deleting a system configuration .....	<a href="#">35</a>
Managing security groups and rules .....	<a href="#">35</a>
Managing users .....	<a href="#">38</a>
Managing the component repository .....	<a href="#">39</a>
Creating and scheduling jobs .....	<a href="#">44</a>
Logging out .....	<a href="#">46</a>
Changing passwords .....	<a href="#">47</a>
<b>Chapter 6: System Monitoring .....</b>	<b><a href="#">49</a></b>
Viewing system status .....	<a href="#">49</a>
Checking log files .....	<a href="#">52</a>
Checking unprocessed records .....	<a href="#">55</a>
Checking system metrics .....	<a href="#">55</a>
<b>Chapter 7: Fine-tuning the DigitalEdge Configuration .....</b>	<b><a href="#">65</a></b>
Adding a new user application .....	<a href="#">65</a>
Configuring an additional data sink .....	<a href="#">65</a>
Data sink parameters .....	<a href="#">66</a>
Sending data to an external application .....	<a href="#">80</a>
Fine-tuning auto-scaling .....	<a href="#">81</a>
Resizing a process or server .....	<a href="#">82</a>
Process group parameters .....	<a href="#">82</a>
Opening a port for a new component .....	<a href="#">84</a>
Assigning IP addresses .....	<a href="#">84</a>
Transport parameters .....	<a href="#">86</a>
<b>Chapter 8: Creating Alerts .....</b>	<b><a href="#">101</a></b>
Building an alerting system .....	<a href="#">102</a>

Specifying alerting criteria .....	<a href="#">102</a>
Managing alert notifications .....	<a href="#">105</a>
<b>Chapter 9: Creating Analytical Dashboards .....</b>	<b><a href="#">109</a></b>
Preparing data for dashboard input .....	<a href="#">111</a>
Creating a dashboard .....	<a href="#">111</a>
Sharing a dashboard .....	<a href="#">112</a>
<b>Chapter 10: Indexing and Searching .....</b>	<b><a href="#">113</a></b>
Configuring an indexing data sink .....	<a href="#">113</a>
Adding a search capability .....	<a href="#">114</a>
Using the search application .....	<a href="#">114</a>
<b>Chapter 11: Troubleshooting .....</b>	<b><a href="#">119</a></b>
Software version .....	<a href="#">120</a>
Accessing applications .....	<a href="#">121</a>
Data Transfer Utility: Connection error when sending data .....	<a href="#">122</a>
How to determine if a service is down .....	<a href="#">122</a>
How to troubleshoot data flow problems .....	<a href="#">123</a>
Ingest or transport process has run out of memory .....	<a href="#">127</a>
Ingest.all is down .....	<a href="#">128</a>
Ingested data is incorrect .....	<a href="#">129</a>
Lucene data sink is full .....	<a href="#">130</a>
Management Console Gateway status is not green .....	<a href="#">131</a>
Management Console Gateway Resources Used are high .....	<a href="#">132</a>
Management Console System Status = Warning .....	<a href="#">133</a>
Management Console System Status = Error .....	<a href="#">133</a>
Management Console: Throttle condition message .....	<a href="#">134</a>
System Builder Error .....	<a href="#">135</a>
System Monitor isn't working .....	<a href="#">135</a>
System Monitor indicates scaling problems .....	<a href="#">136</a>

System Monitor: Can't select a data model for Detail graphs .....	<a href="#">136</a>
System Monitor Storage Utilization graph is blank .....	<a href="#">137</a>
System Monitor, Management Console inconsistent number of instances .....	<a href="#">137</a>
Tableau and the Hive data sink are not communicating .....	<a href="#">138</a>
<b>Appendix A: Terminology .....</b>	<b><a href="#">141</a></b>
<b>Appendix B: What Each Node Does .....</b>	<b><a href="#">145</a></b>
<b>Index .....</b>	<b><a href="#">148</a></b>





## Chapter 1: Introduction

DigitalEdge is a highly configurable software platform providing real-time analytics of big data in motion for cyber-security. This *Operations Guide* helps you with the daily tasks of managing and maintaining the DigitalEdge system after it has been configured.

Before plunging in, be sure to read the *Overview Guide*; it describes the system architecture, concepts, and terminology. These instructions assume you are familiar with those concepts

The *Configuration Guide* helps Data Specialists and DigitalEdge Administrators plan a system, configure data models, build the processing pipeline, and specify user applications. Your team should be finished configuring DigitalEdge before tackling the tasks in this *Operations Guide*.

### Product documentation

DigitalEdge is a complex big data platform. The system comes with a complete set of documentation in PDF and HTML5 formats to help you master DigitalEdge:

Document	Use	Audience
<b>Overview Guide</b>	Basic information about the DigitalEdge platform, including architecture, concepts, and terminology; a must-read before working with any aspect of DigitalEdge	Anyone working with DigitalEdge in any capacity
<b>Configuration Guide</b>	Instructions for defining data models and building processing pipelines	Data Specialists, DigitalEdge Administrators
<b>Operations Guide</b>	Daily administration information, covering monitoring, managing, and modifying the platform	DigitalEdge Administrators
<b>DigitalEdge SDK Guide</b>	Reference for building custom plug-in components	Developers
<b>Alerts API Guide</b>	Reference for specifying data triggers and notifications for an alerting capability	Developers
<b>Search API Guide</b>	Reference for providing search services on a Lucene data sink node	Developers

### Typographical conventions

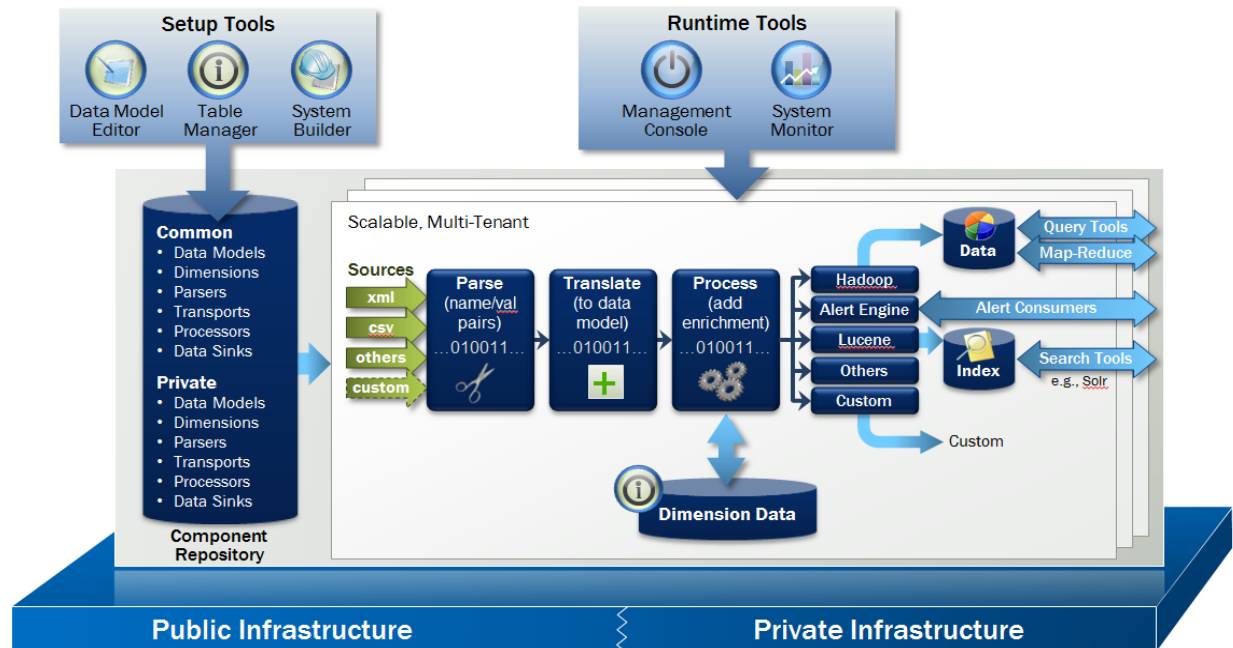
The following style conventions are used throughout this documentation:

Type of Information	Style in Documentation
Code, commands, filenames	<code>code</code>
Cross references	<a href="#">Click to see this topic</a>

Type of Information	Style in Documentation
Emphasis	<i>important point</i>
Hyperlinks	<a href="#">Click to go to this site</a>
Notes, warnings, tips	★
References to other documents	<i>Document Title</i>
Sample code blocks	<code>code</code>
Troubleshooting issue or problem	?
Troubleshooting solution	✓
User input	<i>Italics</i>
User interface labels and controls	<b>Bold</b>
Variables	<code>&lt;change-this-name&gt;</code>

## Chapter 2: DigitalEdge Architecture

DigitalEdge is highly configurable, with a plug-in architecture that lets you swap components in and out. Plug-in components are stored in the Component Repository. The DigitalEdge system architecture is designed as follows:



Data moves through system processors which are configured and customized with the DigitalEdge Setup Tools. System Builder builds and assembles the components into a processing pipeline. The processing pipeline is completely configurable with the Setup Tools. The data flow includes these steps:

1. Transports grab data from data sources and feed the data into DigitalEdge.
2. Data is extracted by parsers.
3. The fusion engine translates and normalizes the data to the DigitalEdge input model.
4. The enrichment engine adds dimensional data and algorithmic enrichments to provide context and meaning to data, resulting in all relevant data being integrated into one record
5. Data is processed and stored in persistent data sinks managed by DigitalEdge (Hadoop, Hive, HBase, MongoDB, etc.) or sent to other data sinks for post-processing (indexing, alerting, etc.). Data can also be sent to systems outside of DigitalEdge.
6. Various web apps makes the data accessible in several ways:
  - Indexed data is searchable through the Search API or the Search app.
  - Configurable situational information is sent to users by the alerting engine.
  - Data can be viewed in dashboards or other external applications.

## Tenant Management System (TMS)

The Tenant Management System is the DigitalEdge application for creating and managing tenant accounts.

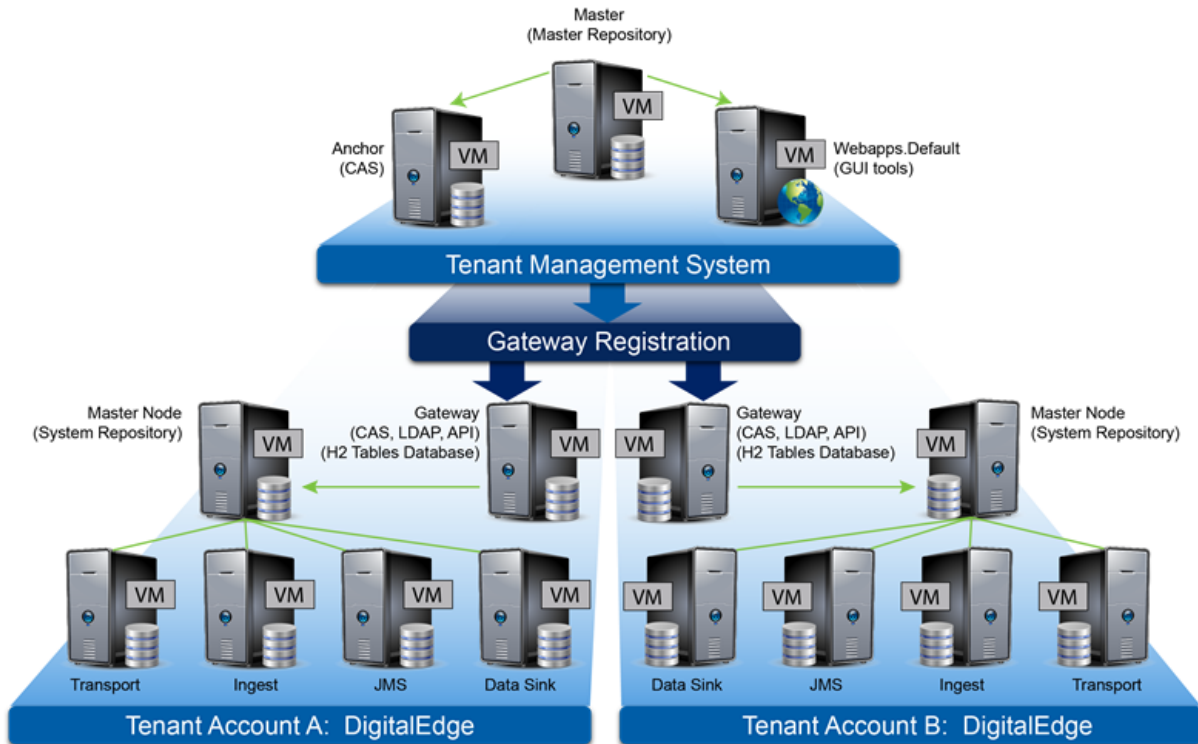
- Public cloud installations: In the AWS™ public cloud, Leidos owns and controls TMS, and each *tenant account* represents an organization that has contracted with Leidos to implement one or more DigitalEdge systems.
- Private cloud installations: On a private Eucalyptus® cloud, TMS is set up at your organization and controlled by your TMS Administrator. A *tenant account* can represent a department, a project, or an organizational group that matches your business needs. Eucalyptus tenants can create independent systems or integrate systems between tenant accounts. They can share plug-in components, and can create private plug-in components as needed.

A *primary tenant* is the first tenant (user) created in an account. The primary tenant owns all the resources: the system repository, LDAP, the tenant database, etc. One or more *secondary tenants* may be created in an account. All secondary tenants share the account resources that are owned by their primary tenant (system repository, LDAP, etc.), share and see all systems created under an account, and have the same privileges as the primary tenant. But secondary tenants have different logon credentials for security purposes. Primary and secondary tenants are created by a TMS Administrator (see the DigitalEdge *Getting Started Guide* for more information.)

Logically, TMS is above the tenant accounts. TMS provides administrative services at an oversight level through the Management Console to:

- Create new tenant accounts
- Manage, set up, and start tenant applications
- Manage user identities
- Store and manage the DigitalEdge private components
- Navigate to other DigitalEdge tools
- View system logs
- Provide an additional level of security

From a high level perspective, TMS and tenant accounts interact as follows:



- TMS is launched at the cloud level, either by AWS™ or Eucalyptus.
- The TMS Master node sets up and launches all the TMS nodes.
- On AWS™, the DigitalEdge Administrator registers with DigitalEdge to configure a new tenant account the DigitalEdge Gateway.
- On Eucalyptus, the TMS Administrator creates new tenant accounts and the DigitalEdge Installer creates the Gateway node.
- The DigitalEdge Administrator builds and starts up DigitalEdge systems.
- The tenant's Gateway node spawns the tenant's Master node for new systems.
- The tenant's Master node launches and manages all other nodes for DigitalEdge systems in the tenant account.

This diagram represents the initialization of a basic DigitalEdge system. Depending on the needs of a tenant's system, the DigitalEdge Administrator may also configure and start up:

- Multiple instances of transport, ingest, JMS, or data sink nodes
- Alerting node(s)
- Search node(s)
- User applications

## About the gateway node

The tenant gateway node hosts CAS for single sign on permissions, all web applications on the account, and LDAP for user account credentials. The gateway node keeps credentials within the tenant account, not in TMS.

The tenant gateway node launches the tenant master node, starts and stops systems, creates and deletes systems and security groups, and synchronizes components.

## About the master node

When configuration activities are complete, you create a new DigitalEdge system simply by running it to start the **Master Node**. The tenant gateway node launches the master node. The master node then handles all instances automatically, monitoring the system for scaling decisions, starting and stopping all instances. Its tasks include:

- Handling virtual storage allocations
- Starting up and shutting down instances in an orderly manner
- Gathering performance metrics as input into scaling decisions
- Adding or removing nodes based on load and storage utilization

The tenant master node also includes the System Repository. In TMS, the master node hosts the Master Repository.

## Chapter 3: Security

DigitalEdge offers a platform built upon a security architecture foundation including account management and provisioning, access controls, logging and auditing, session management, and data security bundled as a service. DigitalEdge includes user and service level identification, authentication, single sign-on, and inter-process session encryption coupled with virtualized host network communication protocol and port restrictions.

DigitalEdge leverages the widely adopted open source JA-SIG Central Authentication Service (CAS) to manage identification and authorization (<http://www.jasig.org/cas>). Coupled with CAS, DigitalEdge is built upon Amazon Web Services™ (AWS™)/Eucalyptus®, leveraging a common set of APIs for management tools across the platforms (<http://www.eucalyptus.com/news/amazon-web-services-and-eucalyptus-partner>). AWS technical, operational, and management security controls secure your system, ensure data privacy, and offer a management platform to build upon. AWS manages the host operating system and virtualization layers down to the physical security of the hosting facilities. AWS has earned certification to operate at the FISMA (the Federal Information Security Management Act) Low and Moderate levels, offering customers visibility for compliance via their Security and Compliance center (<http://aws.amazon.com/security/>). System backup and recovery services are provided at a customized level.

### Certificates

DigitalEdge uses certificates for authentication and encryption, and Secure Sockets Layer (SSL) communication. When a new system is created on DigitalEdge, a self-signed certificate is provided, and is intended only for non-production use. For a production system, the self-signed certificate should be replaced with a wild-carded certificate signed by an industry certification authority.

### Securing data

Data entry and exit points are enabled with SSL over TCP/IP so that the streaming data is encrypted. The system also incorporates bi-directional data certificates between endpoints to ensure data integrity and to prevent man-in-the-middle attacks. If deployed in the Amazon cloud, the system also leverages native Amazon S3 capabilities to support encryption of data at rest.

For AWS™ implementations, the platform can be configured to replicate across multiple AWS zones for backup and disaster recovery. For private cloud solutions, the backup and archival capabilities would leverage additional resources in a parallel environment.

### Security markings

The data modeling of the ingest pipeline can accommodate data labeling, depending upon your DigitalEdge configuration. Security marking can inherit or translate native data markings within the ingest data content (if present). Alternatively, the modeling process can accommodate markings based upon the data source if markings are not native to the data source. When the system does not support all aspects of a required data marking implementation, DigitalEdge provides the flexibility to extend or enhance existing data parsers and enrichments to achieve your objectives.

## Security perimeters

While security precautions have been taken such as port restrictions and instance hardening, it is still important for you to ensure that your DigitalEdge system is behind a robust security perimeter, such as a firewall or other similar network protection device, to prevent sophisticated forms of intrusion.

Use the DigitalEdge Management Console application to modify security group settings and policies, including:

- Opening and closing ports for instances and components
- Defining security roles for security groups
- Defining roles to allow connections to virtual instances
- Associating instances with security groups

The combination of these activities further limits access to the systems.

## Management Console security

The Management Console use HTTPS over TCP/IP to ensure the privacy of network communications containing system settings and configuration.

## Logs

The Management Console lets you view logs captured in the system. Each instance has its own set of log files.

## User auditing

The DigitalEdge Platform uses the open source JA-SIG Central Authentication Service (CAS) system to enable user access and to enforce login policy. The system audits all login attempts, successful or otherwise. The system can also log changes made through the Management Console application, which controls user accounts and privileges for the platform.

## User management and administration

Each account has a DigitalEdge Administrator (the primary tenant account) who manages users through the Management Console. Use the Management Console for:

- Provisioning of user accounts
- Resetting of passwords
- Disabling, re-enabling, and deleting of accounts

These actions can only be done by the Management Console administrative role, not by the general user. General users can set and change their individual passwords, but cannot assign user roles.

## User authentication

The DigitalEdge Platform utilizes user authentication through a CAS interface. CAS provides:



- Enterprise single sign-on
- An open and well-documented protocol
- Numerous libraries, components, and adopters

For details, see <http://www.jasig.org/cas>.



## Chapter 4: Getting Data into the System

### Plug-in components

To get data into DigitalEdge, you must specify the front end transport that will feed data into the parsers. DigitalEdge provides several tools to feed data into the system. For systems on the public Amazon cloud, data is usually uploaded or directed to an S3™ bucket that can be transported to the JMS external nodes. For private cloud systems, other transports may be more useful for grabbing data from a TCP stream, UDP service, file system, database, or a corporate JMS server.

If you have a data stream that has been used locally in a legacy system, the data feed should be diverted to Amazon via Java or JMS. If you have a repository of legacy data that DigitalEdge must process, you can upload the file to an Amazon S3™ bucket, or use any of the local transport methods.

Use the **System Builder** tool to select data transports from a repository of plug-in components.

If the standard DigitalEdge transports do not meet your needs, you can use the **DigitalEdge SDK** to create a custom transport. For example, you may have a continual data feed that must be directed to the DigitalEdge JMS external queue. In that case, the URL transport may not be sufficient, but may serve as the foundation for writing a new transport.

### Data Transfer Utility

DigitalEdge provides an alternative and quick method of getting data into the system which circumvents the front-end transport. The **Data Transfer Utility** (DTU) provides a simplified path for getting data from your local desktop machine to the JMS external queue. Any file that is accessible via Windows Explorer can be ingested by the **Data Transfer Utility**. The DTU was developed as a useful GUI tool for prototyping and testing DigitalEdge data models and sample data sets.

### For more information

See the DigitalEdge *Configuration Guide* for detailed instructions on configuring transports/data sinks/parsers/data models and running a new system.

See the DigitalEdge *SDK Guide* for information about creating a custom transport.

### Getting data in with transports

Each data source must be associated with a transport mode to get data into DigitalEdge. Transports simply move data into DigitalEdge wrapped in JMS messages; they do not convert data to JSON.

Ingest supports several secure automated transport mechanisms, including file-based transfer protocols, streaming TCP and UDP connections, external database queries, and unstructured documents. For example, the TCP transport establishes a TCP socket listening to a port, converts it to JMS messages, and pushes the data into the JMS external node. Each parser can be assigned to a different transport type of your choice. If you need a transport that is not included in the core release, your developers can create a custom transport for your site.

### **DatabaseWatcherTransportService**

The Database Watcher transport is a specialized polling service that gets data from a database and pulls it into DigitalEdge by running an SQL select query against any database. The database can be queried regularly, starting at the point where the query last left off. An S3 bucket is used to store a backup copy of the data file.

### **DirectoryCrawlerTransportService**

The Directory Crawler beta transport is similar to the Directory Watcher transport, processing data in a local or remote file system. But it also decompresses zipped files and processes files that match wild card patterns.

### **DirectoryWatcherTransportService**

This transport is a polling service similar to the PollingS3TransportService. This transport watches a local or remote file system rather than an S3™ bucket. Use the Directory Watcher transport when you have data files that you do not want to move to S3.

### **HiveTransportService**

The Hive transport is a specialized transport that gets data from an existing Hive data sink and pulls it into another data sink, either in the same DigitalEdge system or another DigitalEdge system. For example, you might store enriched data in Hive and then transport it to a Lucene data sink for indexing. You can optionally create an SQL select query to run against Hive to pull out a subset of data.

### **JMSBridgeTransportService**

This service copies data directly from your corporate JMS server to the DigitalEdge JMS server without a transport. If your shop has multiple JMS brokers, this is probably the easiest service to use. The JMS Bridge Service pushes data directly onto a JMS queue without a transport. For example, there could be two JMS servers in play: a corporate JMS server and the DigitalEdge JMS server; the bridge pushes data from your enterprise queue to DigitalEdge. If you do not use JMS internally, you should choose another transport from the DigitalEdge repository.

### **MongoDBTransportService**

The MongoDB transport is a specialized transport that gets data from an existing MongoDB data sink and pulls it into another data sink, either in the same DigitalEdge system or another DigitalEdge system. For example, you might store enriched data in MongoDB and then transport it to a Lucene data sink for indexing. You can optionally create an SQL select statement to run against MongoDB to pull out just a subset of data.

### **PcapSnifferTransportService**

This transport captures and splits pcap packets on a specific network interface. You can optionally filter out data and pull selected relevant data. The pcap transport is often used with the SNMP PCAP parser.

### **S3FileTransportService**

The S3 transport service is a good choice when you have one or more static files, such as legacy data, to get into DigitalEdge. This S3 transport can be configured two ways. As a one shot event, it pulls data from an Amazon's S3™ (Simple Storage Service™) file and pushes it to the DigitalEdge JMS queue. Or, you can configure the transport to poll an S3™ bucket regularly at a set time interval. Choose this option when you have an existing system that is generating large files and you are adding DigitalEdge to the system as the big data processor and analyzer. Set up buckets for data feeds from several data sources, one bucket per data type. Use the polling transport to check the bucket for new data every several minutes, hours, or once a day, depending on your time-to-availability requirements. You can configure this transport to either save the file or delete it immediately after processing.

### **TCPTransportService**

This is a commonly used transport, to read data in a TCP stream. The TCP socket listens to a port, converts the data to a JMS message, and puts it on the JMS external queue.

### **TwitterFilterTransportService**

This transport gets Tweets from the Twitter feed based on criteria that you define via the Twitter Search API. You can search for keywords and/or Twitter usernames. This is the most flexible and commonly used transport of the three Twitter transports.

### **TwitterRESTTransportService**

This transport follows the Tweets of one Twitter user. The transport uses the Twitter REST API.

### **TwitterSampleTransportService**

This transport selects a random sample of Tweets that you are allowed to read in the Twitter feed. The transport uses the Twitter Search API.

### **UDPTransportService**

This frequently used transport captures network packets in a UDP stream, converts the data to JMS messages, and sends them to the JMS external node.

### **URLTransportService**

This transport reads the contents of a URL and puts it on the JMS input queue. Use this transport to pull data from an RSS feed or from any service that pulls resources from another organization or data source.

## **Transport examples**

Amazon's S3™ service (Simple Storage Service™) is a cost-effective way to store all types of data for applications which require a resilient, scalable, and accessible medium for data exchange. Many

companies use S3 for a wide range of applications, such as e-commerce, scientific computing, video/audio processing, financial data, etc.

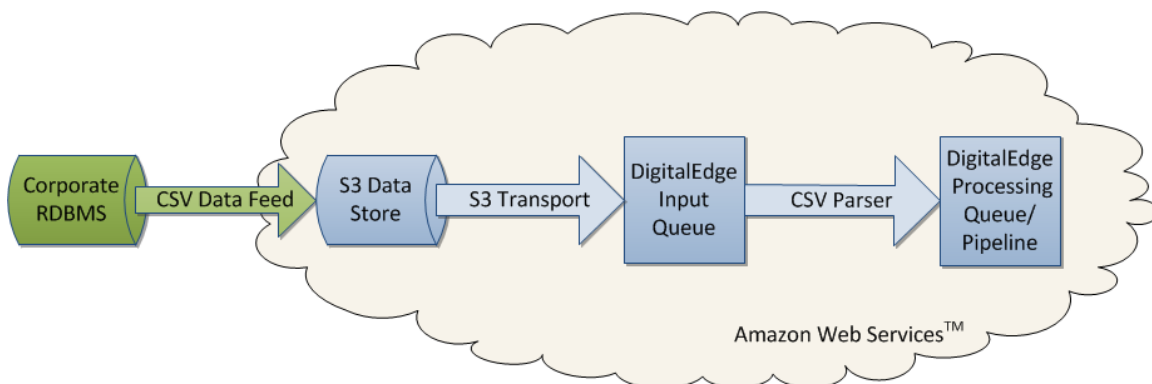
DigitalEdge includes the S3 File Transport which processes data in user account S3™ buckets. The S3 transport service is a good choice when you have one or more static files, such as legacy data, to get into DigitalEdge. The S3 transport can be configured two different ways:

- To read the S3™ bucket once - As soon as the file appears in the bucket, DigitalEdge pushes the file to the JMS queue.
- To poll the S3™ bucket periodically - The transport checks the bucket regularly at a set time interval. It may locate multiple files over time with the same name. This is a good option to use when you have an existing system that is generating large files and you are adding DigitalEdge to the system as the big data processor and analyzer. You can configure the transport to check the bucket for new data every several minutes, hours, or once a day, depending on your time-to-availability requirements.

Using S3 as a data sharing medium makes it easy for organizations to integrate DigitalEdge into existing systems and to leverage DigitalEdge's processing capabilities. You do not have to use a proprietary API to create a transport, you can quickly get data into DigitalEdge, and you do not have to move data to a new database or repository. DigitalEdge can be integrated into an existing system with very little effort, offering new and innovative ways to use your data and to create innovative analytical solutions.

### Relational database export processing

An organization might use an S3™ transport to add additional capabilities to an existing stack without migrating or replacing existing solutions. If you have a way to produce a consumable data stream, you can simply route a portion of the feed to an S3™ bucket. For example, you could create a scheduled job to produce CSV exports from your application's RDBMS and to feed the data into DigitalEdge. The CSV exports could be copied to S3, using existing capabilities or Open Source transfer solutions. Use the S3FileTransportService to migrate the constant stream of data into S3™ buckets for periodic DigitalEdge processing.



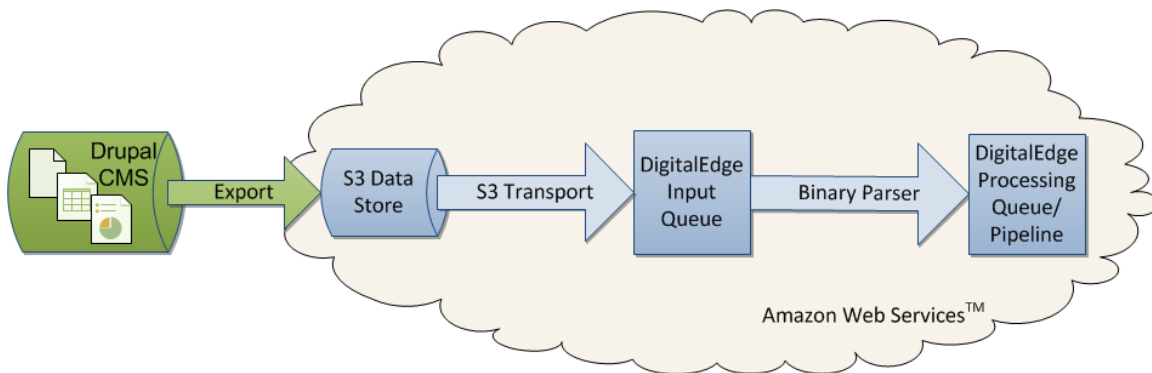
## Non-CSV processing

Although traditionally, CSV exports are a common mechanism for data exchange, the DigitalEdge S3 transport can be used to process other forms of data with little effort. You can use S3™ buckets to process:

- traditional office files such as word processor documents, PDFs, spreadsheets, etc.
- audio or video data streamed in from an existing enterprise technology such as share drives, CMS systems, email servers, etc.

While it may require a modest investment of time and effort to hook into existing platforms and create a stream or copy of the data to S3™ buckets, once accomplished, DigitalEdge processing will be transparent in its application against a very large data set.

For example, you could leverage DigitalEdge to perform copyright infringement analysis on a corpus of intellectual proprietary data compared to a set of data from the competition. The data may be stored on a CMS system such as Drupal®, with a stream or copy placed somewhere accessible by DigitalEdge's processing pipeline. Since exporting Word and PDF documents from Drupal is a trivial task, you would perform a one-time export to S3 where the DigitalEdge S3FileTransportService would find and process the data.



## Getting unstructured data in

DigitalEdge is equally proficient at ingesting and processing unstructured data as well as structured data. A data model does not have to be overly complex and extensive to be effective. Unstructured data such as email messages, Word documents, and PDFs can be captured and available for searching and analysis with or without extracting properties and metadata.

Use the following guidelines when you need to:

- Create a quick prototype of a new system without specifying a complete set of input fields
- Create a simple system for data discovery, so you can evaluate record content that you are not yet familiar with
- Index and search unstructured data that is not fielded and/or contains metadata inside a complex file structure

Here are the plug-in components that you will use to build a simple unstructured data system:

- A data model that includes just 2 fields
- Unstructured File Parser
- Lucene Data Sink: for indexing and searching the data
- Directory Crawler Transport: to ingest local files
- Search API webapp: to do federated searches
- Search webapp: to do quick searches in a GUI

## Build the system

The strategy for building a simple prototype system for unstructured data is to extract all the source file content and metadata, and to process it all into just one or a few DigitalEdge fields for output.

1. In the **Data Model Editor**, create a data model that includes two string type fields: content and filename.
2. When you define your data source, select the **UnstructuredFileParser**. This parser can split the unstructured data into as many or as few fields as you need.
  - a. Initially, don't define any input fields for the parser.
  - b. Accept the defaults for field mapping.
  - c. For the extracted-content-length parameter, start with the default value of 250 MB. This parameter applies to text only and does not affect embedded graphics.
  - d. The parser will automatically preserve the filenames of the incoming data source files.
3. In **System Builder**, select the **DirectoryCrawlerTransportService**.
  - a. Create file mappings with the map-input-models parameter.
  - b. Set the transport's content-encoding parameter to `Base64`.
  - c. Set the record-format parameter to `NULL`.
4. In **System Builder**, select the **LuceneIndexingDataSink** and accept all its default parameter values to start with.
5. In **System Builder**, select the following webapps/REST APIs:
  - a. **search**: for a simply query interface
  - b. **searchapi**: to perform federated queries
6. **Build** the system

## Search the records for support in data discovery

Once the system is built and the data is indexed in the Lucene data sink, you can examine and explore the unstructured data for patterns, potential metadata, and key content.


1. In the **Management Console**, **Start** your system.
2. When the system status is **OK**, select the system and look at the **Process Groups** tab in the bottom panel.



- a. Double-click the **webapps.main** group. This is the instance where your search apps run.
- b. Get the **Public IP** address for the search interface.
- c. Open the IP in a browser to access the search interface.
- d. The **JSON Record** representation shows the content for each unstructured record. Click **More** to see the full record.
- e. Use this as a first step in your data discovery process to decide if you want to extract, define, and index content fields.
- f. Search for any key content that you are looking for.

### Mine the log files

Next, consider examining the ingest log files to discover potential metadata fields for indexing.

1. In the **Management Console**, select your system and click the View Logs icon .
2. Expand **ingest.all** and examine the numbered log file(s).
3. Scan through the file for any **Detected Metadata Key** that may be a useful property from Office or PDF documents.
4. You can then refine your data model in the **Data Model Editor** by doing a **Get Field** on any property that should be added to your input model for indexing.

### Getting data in with the Data Transfer Utility

The **Data Transfer Utility** (DTU) provides a simplified path for getting data from a local computer to the JMS external queue by acting as a *local* directory crawler transport to ingest files from your desktop or from a mapped network drive (any file that is accessible via Windows Explorer). Rather than waiting for a scheduled job or a transport to process a web-based data stream, you control when to run the DTU on a *local* data subset. It is highly effective for getting sample data sets into DigitalEdge test and prototype systems. The Data Transfer Utility uses SSL to transport data securely into the JMS external node and uses available memory to ingest data.

For large production systems, you should specify a complete processing pipeline with **System Builder** rather than using the DTU as a desktop transport mechanism. Consult the *DigitalEdge Configuration Guide* for detailed instructions on configuring transports/data sinks/parsers/data models.



Before using the DTU, you must create an input data model either by doing a full configuration with the Data Model Editor, or by using the Data Model Creation Wizard to construct a simple data model from a flat file. You will need a data model, data sources, and data field mappings so the DTU can transport your data to a DigitalEdge processing pipeline. See the *DigitalEdge Configuration Guide* for details.

---





Your DigitalEdge system must be **Started** and running in the **OK** state before using the Data Transfer Utility.

---

The DTU can be used as a GUI application or through its command line interface.

## Downloading the Data Transfer Utility

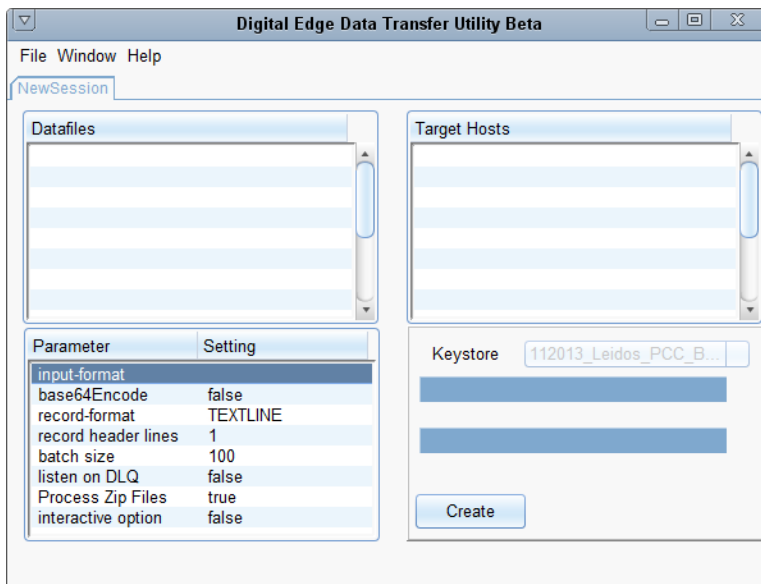
The DTU application is an executable JAR file, downloaded from the DigitalEdge**Management Console**.

1. From the **Management Console**, click the **Help** icon  to access the **Product Documentation** page.
2. On the **Data Transfer Utility** line, click Download .
3. Click **Save File** to copy the DTU file to a directory with read/write access.
4. Click on the executable JAR file to run the DTU.

## Using the Data Transfer Utility GUI

Use the GUI to define a data transfer job by specifying input files, destination hosts, keystores, record formats, and other options. You can create and save a transfer job as a "session" in the DTU to run immediately or repeatedly over time.

1. Open the Data Transfer Utility:



2. To create a new data transfer job, simply enter information into the **NewSession** tab.
3. In the left **Datafiles** panel, right-click anywhere to access the **Add/Delete** dialog box. Click **Add Row** to access the **Open** dialog box. Browse to select an input source data file or a directory. To use multiple input files in one data transfer session, repeat this process to add files or directories, one at a time. Or enter the data directly in the **Datafiles** panel without using the **Open** dialog box.



If you specify a directory as the input, DTU will recurse into all subfolders.

- 
- ★ You can use wildcards in filename specifications (? = one character; \* = multiple characters).
- 
- ★ To edit a file name, double-click on its row.
- 
- ★ To delete a file, right-click on a row to access the **Add/Delete** dialog box. Click **Delete Row**.
- 

4. Specify the data model parameters for this job:
  - **input-format**: Specifies the data source that DTU uses to pull data off the input queue; enter as the Data Source name from the Data Model Editor. This parameter also identifies the data model and field mappings that are associated with this data source and which DigitalEdge will use to process your data once it is transported into DigitalEdge by the DTU (DTU does not do data modeling or field mapping).
  - **base64Encode**: Click to select true or false; indicates if the resulting records should be encoded in Base64; use true only with the Binary Parser. Default = false
  - **record-format**: Specifies a data record type that helps to determine record boundaries in the source data. Click to select a type of record splitter to use: PCAP, JSON, TEXTLINE, NULL, or TEXTLINEWITHQUOTES. Default = TEXTLINE
  - **record header lines**: The number of lines of header information in the source file(s); enter 0 if no header is present. Default = 1
  - **batch size**: Defines the size of the destination JMS message batches. Default = 100 messages
  - **listen on DLQ**: Click to select true or false; indicates if DTU should listen for messages on the Dead Letter Queue (DLQ). Messages/records go to the DLQ when they cannot be parsed. If this parameter value is true, the DTU will not self-terminate when the transfer session is completed, but will listen on the DLQ and output messages until you kill the DTU. Default = false
  - **Process Zip Files**: Unzips and processes zipped files: ZIP, TAR, TAR.GZ, TGZ, TAR.BZ2, TBZ2, GZ, 7Z, RAR, BZ2. Default = true
  - **interactive option**: Click to select true or false; Indicates if the DTU should pause between the processing of each JMS message. Default = false
5. In the right **Target Hosts** panel, right-click anywhere to access the **JMS Queue Selection** dialog box. Enter an IP address or DNS name for the target JMS external node that will receive the input data. You can find the IP address at: **Management Console > Systems > Process Groups > jms.external** > double-click for the IP addresses (use the **Copy** function to copy an IP address that you want to paste here). To use multiple queues in one data transfer session, repeat this process to add IP addresses or DNS names, one at a time. (Or enter the data directly in the **Target Hosts** panel without using the **JMS Queue Selection** dialog box.)
6. Select a keystore to encrypt the connection from your PC to the JMS node. Recommendation: Use the default keystore as both a keystore and truststore.



If you do not have a keystore, you can generate and download a keystore from the Management Console (**Management Console > Security > Download > Keystore**). Save the file in a keystore subdirectory under the DTU installation directory.

7. Save your data transfer specifications in a "session" by clicking the **Create** button. DTU saves your session and names it sequentially (Session\_0, Session\_1, Session\_2, etc.).
8. Once you have all options specified, you should test the connection before running the data transfer job.
  - a. Click **Test** to verify the connection, SSL, and the existence of the specified files.
  - b. Any warnings or errors will be reported above the **Test** button and saved in the **Logs** file.
  - c. Click the **Logs** tab (or select **Window > Logs**) to review the text results.
  - d. Correct any problems and retest the connection.
9. To run the session, first be sure that your DigitalEdge system is started and running in the Management Console. Click the **Session\_n** tab here, then click **Upload**. When the session runs, the input files are uploaded to the JMS queue and processed.
  - a. Status is indicated by the two progress bars above the **Test** button.
  - b. The first progress bar indicates overall progress.
  - c. The second progress bar tracks the transmittal of individual files.
  - d. Processing, warnings, and errors are captured in the **Logs** file. Click the **Logs** tab (or select **Window > Logs**) to review the session's history.
  - e. If a record fails to be processed, it is sent to the Dead Letter Queue (DLQ) and recorded in the **Logs** file. Click the **DLQ** tab (or select **Window > DLQ**) to view records that ended up in the DLQ. You can also copy records from the DLQ to another file, edit them, and resubmit them to DigitalEdge.

## Managing DTU sessions

Here is a quick reference list for using DTU functionality.

Data Transfer Utility: GUI Functionality	
Task	GUI Function
Create a new session (data transfer job)	<b>NewSession</b> tab
Run a session (push data files to the JMS queue)	<b>Upload</b>
Stop a running session	Close a <b>Session</b> tab, or <b>Cancel</b> a session, or <b>File &gt; Exit</b>
Restore a previous session	<b>File &gt; Open</b>

Data Transfer Utility: GUI Functionality	
Task	GUI Function
Create a new session from a similar session	<ol style="list-style-type: none"> <li>1. <b>File &gt; Open</b>. Select a previously saved session.</li> <li>2. Edit the previous session (with new parameters or different data files)</li> <li>3. The session is auto-saved with the session number and a timestamp as a filename.</li> </ol>
View DTU processing logs	<b>Logs</b> tab, or <b>Window &gt; Logs</b>
View the Dead Letter Queue (records that were not parsed)	<b>DLQ</b> tab, or <b>Window &gt; DLQ</b>
Read DTU help instructions	<b>Help &gt; Manual (PDF)</b>
Save a session	Click <b>Create</b> for DTU to number your session (Session_1, Session_2, etc.)

### Using the Data Transfer Utility command line interface

The Data Transfer Utility (DTU) also provides a command line interface which can be called from any location. Use the following syntax:

```
Java -jar DataTransferUtility.jar <arguments>
```

If no parameters are passed in, the DTU will run in GUI mode. Use the `-?`, `-a`, and `-m` parameters for information only; all other parameters will run the DTU. For example:

```
Java -jar DataTransferUtility.jar -d 192.160.50.100
-k C:\Users\Smith\Downloads --trustStore C:\Downloads
-f C:\Users\Smith\Documents\export.json
-s JSON -i mydata -l -z 100 -h 0
```

Here are the parameters available to the DTU, along with descriptions and arguments.

Data Transfer Utility: Command Line Interface		
Parameter	Shortcut	Explanation
<code>--help</code>	<code>-?</code>	Displays a list of DTU parameters and explanations
<code>--availableRecordFormat</code>	<code>-a</code>	Displays a list of the available record format types (for input with the <code>--recordFormat</code> parameter); when you use this parameter, DTU ignores all other parameters

Data Transfer Utility: Command Line Interface		
Parameter	Shortcut	Explanation
--encode	-b	Use Base64 to encode the resulting records; used only with the Binary Parser
--testConnection	-c	Tests the connections and files but does not send any JMS messages yet
--destination <arg>	-d	List of comma-separated public IP addresses or DNS names for the target JMS external nodes
--files <arg>	-f	Path to the input source file or files (directory) that will be sent to JMS.
--recordHeaderLines <arg>	-h	Number of lines of header information in the source file(s); specify 0 if no header is present
--inputFormat <arg>	-i	Identifies the source and parser format that DTU uses to pull data off the input queue; specified as the Data Source name from the Data Model Editor
--keyStore <arg>	-k	Path to the keystore that encrypts the connection from your PC to the JMS node; you can obtain a keystore in the Management Console <b>(Management Console &gt; Security &gt; Download &gt; Keystore)</b>
--listen	-l	Listens for messages on the Dead Letter Queue (DLQ). Messages go to the DLQ when they cannot be parsed. If this flag is specified, the DTU will not self-terminate when completed, but will listen on the DLQ and output messages until you kill the DTU.
--md5	-m	Calculates the MD5 hash of the DTU file to verify that the file has not been tampered with; when you use this parameter, DTU ignores all other parameters
--zip	-p	Unzips and processes zipped files: ZIP, TAR, TAR.GZ, TGZ, TAR.BZ2, TBZ2, GZ, 7Z, RAR, BZ2
--recursive	-r	Recurses into subfolders if the input is a directory (specified in the --files parameter)
--recordFormat <arg>	-s	Specifies a record format type that determines record boundaries in the source data; used to split the data stream into logical records; to see the complete list of available record types, use --a
--trustStore <arg>	-t	Path to the truststore that authenticates the connection between your PC and the JMS node; you

Data Transfer Utility: Command Line Interface		
Parameter	Shortcut	Explanation
		can obtain a truststore in the Management Console ( <b>Management Console &gt; Security &gt; Download &gt; Truststore</b> )
--interactive	-w	Pauses between JMS message batches (see --z) and files to review processing for each message; waits for input before proceeding
--mapInputModels	-x	<p>A CSV string that specifies input model mapping parameters, including a Data Source name, the record format for splitting records, and the number of header lines. When you use this parameter, DTU ignores the individual parameters:</p> <ul style="list-style-type: none"> <li>--recordFormat</li> <li>--inputFormat</li> <li>--recordHeaderLines</li> </ul> <p>Specify file mappings in the following format:</p> <pre>fileName1:recordFormat1:recordHeaderLines1: inputFormat1,fileName2:recordFormat1: recordHeaderLines2:inputFormat2,fileName3: recordFormat3:recordHeaderLines3:inputFormat1</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• fileName can include wildcards: <ul style="list-style-type: none"> <li>◦ ? represents a single character</li> <li>◦ * represents multiple characters</li> </ul> </li> <li>• recordFormat is the data record type to determine how to split multiple records on the appropriate boundaries</li> <li>• recordHeaderLines is the number of header lines that can be stripped out of the records</li> <li>• inputFormat must match a Data Source name as defined in the Data Model Editor</li> </ul> <p>White spaces are ignored.</p>
--batchSize <arg>	-z	Sets the size of the JMS message batches; default = 100 messages



You can also embed DTU commands and parameters in a script to run from your local machine.

---

## Generating metrics for CSV files

DigitalEdge provides the ability to profile ingested CSV files to find potential data problems and to perform an audit on the data. The profiling generates metrics on errors found in the data, which can help determine how valuable or valid the data is before it is processed.

Each processed file will contain its own set of metrics. The following metrics are gathered by the profiling process:

- The total number of records processed
- Records that failed to be parsed by the DigitalEdge CSV Parser component
- Records in which the field count does not match the file header
- Records with non-ASCII characters
- Records with empty fields

A single record can generate multiple metrics, based on the above list. For example, a record could contain both non-ASCII characters and empty fields, and would be counted in the metrics as both having non-ASCII characters and empty fields.

## Configure the system(s)

To generate metrics on CSV files, first prepare your DigitalEdge system by configuring the following components in System Builder.

1. Data Model: Choose the Common data model called `csv_parser_metrics_v1.0`

Use the `CsvParser` and set the `mode` parameter to `Metrics`.

2. Transports: Choose one of the following transports to use when generating CSV metrics.

- `DirectoryCrawlerTransportService`
- `DirectoryWatcherTransportService`
- `S3FileTransportService`

When editing the transport parameters, set the `input-format` parameter to `CSV_Metrics`.

3. Data sinks: Use the `MongoDbDataSink`

- Set the `database-name` parameter to `csvMetrics`.
- Leave all other parameters to their default values.



For any parameter that is not listed here, choose any value that is appropriate.

---





If you want to parse/extract CSV fields while also generating metrics on the CSV data, you must create two DigitalEdge systems. One system should use the CsvParser with the mode parameter set to `Metrics`. The second system should use the CsvParser with the mode parameter set to `Parse`.

---

### Ingest the CSV files

When the configured system is running, ingest all the CSV files into the DigitalEdge system. The ingest procedure will vary depending on the transport that you use. Consult the *DigitalEdge Configuration Guide* for detailed instructions about the parameter settings for your selected transport. As the files are processed, DigitalEdge will generate the metrics from the CSV file content.

### Install UMongo

To view the CSV ingest metrics, use UMongo, a GUI application for browsing the MongoDB data sink.

To download and install UMongo:

1. Go to <http://edgytech.com/umongo/>
2. Download the appropriate zip file for your OS.
3. Unzip the contents of the zip file to your computer.
4. In the unzipped folder, select `umongo.jar` to launch UMongo.

### Access UMongo

Connect to the DigitalEdge system's MongoDB data sink with the following steps.

1. Go to **File > Connect** and click the **Edit** button in the **Connect** window. The **Connect** dialog box appears.
2. In the **Servers** field, enter:

```
<Public IP address>:27017
```

where `<Public IP address>` is the public IP of the `mongodb.standalone` instance on the running DigitalEdge system.

3. In the **databases** field, verify that the name matches your MongoDB database name, `csvMetrics`.
4. Expand the **csvMetrics** database in the **Mongo Instances** view. Highlight the **csv\_parser\_metrics** option.

### View the file metrics

You are now ready to review the CSV file metrics that DigitalEdge collected and compiled.

1. In the upper right pane of the UMongo client, select **Command > Aggregate**. The **Aggregate** dialog box appears.
2. Add a new operation by clicking the **+** button.
3. In the **Add Operation** dialog box, select **group** from the drop-down box and click **Ok**. The **Edit Agg Group** dialog box appears.
4. In the text field, paste the following:

```
{ _id : "$filename", totalRecordCount : {$sum :
"$metrics.totalRecordCount"}, totalCsvParserErrorCount : {$sum :
"$metrics.csvParserErrorCount"}, totalIncorrectFieldToColumnCountErrorCount
: {$sum : "$metrics.incorrectFieldToColumnCountErrorCount"},
totalNonAsciiErrorCount : {$sum : "$metrics.nonAsciiErrorCount"},
totalEmptyFieldErrorCount : {$sum : "$metrics.emptyFieldErrorCount"} }
```

5. Click **Ok** to close the **Edit Agg Group** dialog box.
6. Add a new operation by clicking the **+** button.
7. In the **Add Operation** dialog box, select **project** from the drop-down box and click **Ok**. The **Edit Agg Project** dialog box opens.
8. In the text field, paste the following:

```
{ "totalRecordCount" : 1 , "totalCsvParserErrorCount" : 1 ,
"totalIncorrectFieldToColumnCountErrorCount" : 1 ,
"totalNonAsciiErrorCount" : 1 , "totalEmptyFieldErrorCount" : 1 ,
"percentCsvParserErrorCount" : { "$divide" : [ "$totalCsvParserErrorCount"
, "$totalRecordCount"] } , "percentIncorrectFieldToColumnCountErrorCount" :
{ "$divide" : [ "$totalIncorrectFieldToColumnCountErrorCount"
, "$totalRecordCount"] } , "percentNonAsciiErrorCount" : { "$divide" :
[ "$totalNonAsciiErrorCount" , "$totalRecordCount"] } ,
"percentEmptyFieldErrorCount" : { "$divide" :
[ "$totalEmptyFieldErrorCount" , "$totalRecordCount"] }}
```

9. Click **Ok** to close the **Edit Agg Project** dialog box.
10. Click **Ok** to close the **Aggregate** dialog box. The query will now run.
11. The **result** row in the bottom right pane will contain the metrics for each file.
12. The row can be further explored by expanding it to show each record in the result. The **\_id** field of each record is the corresponding file for which the metrics were gathered.

### View errors by file

In addition to the CSV metrics, you can review error reports for records that generated the metrics.

1. In the upper right pane of the UMongo client, select **Document > Find**
2. In the **Query** text field, paste the following:

```
{"filename":"<searchFileName>","errorMessages":{"$not: {$size: 0}}}
```

where **<searchFileName>** is replaced with the file you are searching for.

3. To limit the number of results, set the **Limit** field to the desired number.
4. Click **Ok** to run the query. The resulting fields will appear in the bottom right pane. The results will contain:
  - **filename**: the file for which the errors occurred
  - **metrics**: the metrics accumulated since the file start or the last seen error
  - **recordHeaders**: the file's headers
  - **record**: the fields which generated the error
  - **errorMessages**: the reasons that errors were generated



## Chapter 5: Managing DigitalEdge from the Management Console

Once you have a system configured, built, and ready to test or to run in production, the **Management Console** is the runtime tool that you will use every day. The Management Console is the main portal into DigitalEdge, providing access to all the major UI tools. The Management Console dashboard includes icons for starting and stopping systems. Status views in both the Management Console and the System Monitor support assessing and troubleshooting system health and performance.

The screenshot shows the DigitalEdge Management Console interface. The top navigation bar includes links for Management Console, System Builder, Data Model Editor, Table Manager, and System Monitor. The left sidebar contains a 'Tools' menu with icons for Systems, Users, Security, Plug-ins, System Builder, Data Model Editor, Table Manager, and System Monitor. The main content area displays a table of systems with columns for System, Status, VPC ID, Subnet ID, # Instances, # Volumes, and Controls. The 'Controls' column contains icons for starting, stopping, and deleting systems. Annotations with red arrows point to various features: 'Service zone working in' points to the top left; 'Manage, start, and stop systems' points to the Systems icon; 'Create user accounts for accessing Web applications' points to the Users icon; 'Maintain security groups and rules' points to the Security icon; 'Upload private components and download common components' points to the Plug-ins icon; 'Access Setup and Runtime Tools' points to the System Builder icon; 'View a snapshot of system resources' points to the Table Manager icon; 'Delete old system configurations' points to the System Monitor icon; 'Create and run scheduled jobs' points to the System Monitor icon; 'Access system metrics and status reports' points to the System Monitor icon; 'Jump to any Setup or Runtime Tool' points to the top navigation bar; 'Access product documentation and help files' points to the top right; 'Log out' points to the top right; 'Access log files' points to a log icon in the Controls column; 'Access System Monitor' points to a monitor icon in the Controls column; 'Update a system' points to a refresh icon in the Controls column; 'Start a system' points to a power icon in the Controls column; 'Stop a system' points to a stop icon in the Controls column; and 'Delete a system' points to a delete icon in the Controls column.

Use the Management Console to:

- **Start a new system:** [See "Starting a system for the first time" on page 30](#)
- **Stop and restart a running system:** [See "Shutting down and restarting a system" on page 31](#)
- **Stop and restart an individual process group:** [See "Stopping and restarting a process group" on page 32](#)
- **Update a system with edits to a data model, transport parameters, or data sink parameters:** [See "Updating a system" on page 33](#)
- **Delete a system:** [See "Deleting a system" on page 34](#)
- **Delete a system configuration:** [See "Deleting a system configuration" on page 35"](#)
- **Maintain security rules for security groups:** [See "Managing security groups and rules" on page 35](#)

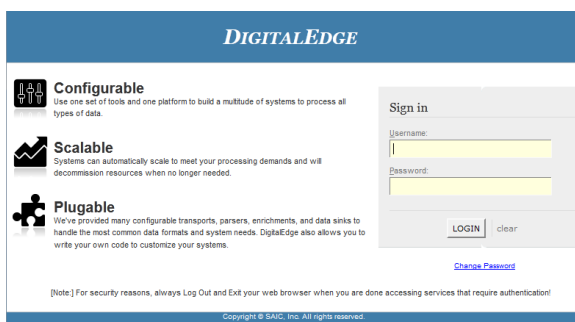
- **Create and maintain user accounts** for accessing web apps: [See "Managing users" on page 38](#)
- **Get custom-built plug-in components into the system:** [See "Managing the component repository" on page 39](#)
- **View a quick snapshot of the system's status:** [See "Viewing system status" on page 49](#) (to edit process group parameters in **System Builder**, [See "Process group parameters" on page 82](#))
- **View log files:** [See "Checking log files" on page 52](#)
- **View system metrics:** [See "Checking system metrics" on page 55](#)
- **Access all the setup and runtime tools** (Data Model Editor, Table Manager, System Builder, and System Monitor) from icons on one convenient **Management Console** dashboard:

**NOTE:** Each Setup and Runtime tool will time out after approximately 30 minutes of inactivity. Also, if you have multiple tools open on separate tabs of a web browser, all the tools will time out if one tool reaches the timeout threshold. On the timeout screen, you can **Sign Out** of your session (and lose unsaved data), **Sign Back In** to re-enter any unsaved data, or **Work Offline** to record any work you were doing that was not saved.

## Logging in

Use this procedure to log on to the DigitalEdge **Management Console**.

1. In a web browser, go to `https://default.<system_domain_name>/tenantconsole`
2. Enter your **Username** and **Password**.
3. Click **LOGIN**.



You can access all the DigitalEdge Setup and Runtime tools from the **Management Console**.

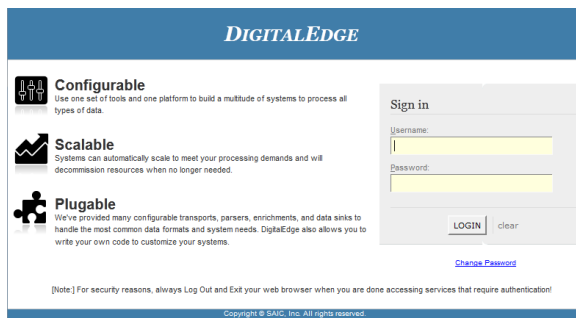


You cannot access the **Management Console** with an expired DigitalEdge license; contact Support for a new license.

## Starting a system for the first time




To start a new system for the first time:

1. Open the **Management Console**.
  - a. In a web browser, go to `https://<system_domain_name>/tenantconsole`



- b. Enter your **Username** and **Password**.
  - c. Click **LOGIN**.

★ The first time you access the **Management Console**, a **License Agreement** will appear. Read the agreement and click **Accept** to continue.

1. From the **Systems**  **Systems**  screen, highlight the **System** name that you want to launch.
2. Click the **Start System** icon.  The **Select a version** dialog box appears.
3. Click the **Version** number you want to run and click **START**.

★ Starting a system can take some time. It takes 1-3 minutes to launch each instance in a system.




4. The **Status** of the newly started system changes in the **Systems** list. [See "Viewing system status" on page 49](#) for an explanation of each status.


## Shutting down and restarting a system

There are times when you have to stop your system temporarily then restart it. Stop a system when you:

- Add a new plug-in component (you cannot overwrite the system configuration of a system that is running)
- Reconfigure any part of the system (you cannot overwrite the system configuration of a system that is running)

In the **Management Console**:

1. Click the **Systems** option.  **Systems** 
2. Click the **System** name you want to work with.
3. Click  **Stop System** to temporarily stop a system.

4. You can restart the system with the **Start System** icon. 

---

★ Stopping or starting a system can take some time. Expect approximately 1-3 minutes per instance in a system.

---

★ You can also stop and restart select process groups instead of stopping the entire system. [See "Stopping and restarting a process group" on page 32](#)

---

★ You can update a running system after editing a data model, transport parameters, or data sink parameters. [See "Updating a system" on page 33](#)

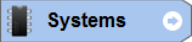
---

## Stopping and restarting a process group

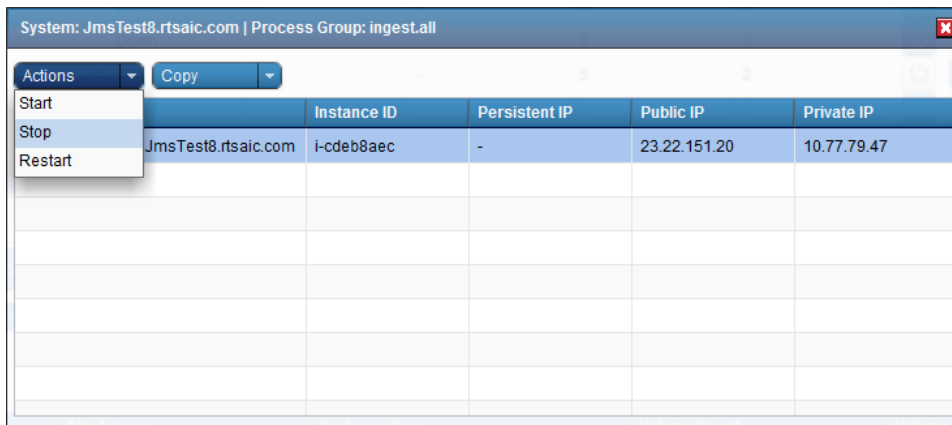
There are times when you need to stop and restart one process group. You can do this for a **transport**, **ingest.all**, or **datasink.lucene** without stopping the entire system. For example, you may need to stop and restart a process group in these situations:

- The ingest.all node is down and the log viewer indicates a minor problem. Bring up the ingest process by Restarting that process group.
- Data entering the system is incorrect. The Systems Administrator Stops ingest.all to examine and correct the data before Restarting ingest.
- The Lucene data sink has filled up and you want to wipe out the data to start fresh with new data. Purge the data sink and start it up again.
- You need to stop the flow of data by Stopping the transport process group.
- The transport, ingest.all, or datasink.lucene process has run out of memory. To get it back into a working state, Restart that process group.

In the **Management Console**:

1. Click the **Systems** option. 
2. Click the **System** name you want to work with.
3. On the **Process Groups** tab, double-click the individual **Group Name** you wish to stop: **transport**, **ingest.all**, or **datasink.lucene**. A System dialog box appears.
4. Highlight the name of one or several process groups.
5. On the **Actions** menu, select **Start**, **Stop**, or **Restart** (**Restart** is a combination of **Stop** and **Start**).





★ Note that when working with the **datasink.lucene** process group, the **Actions** menu includes an additional choice: **Purge Data**, which wipes out the old index and automatically **Restarts** the indexing process.


6. DigitalEdge processes your request and reports when the action is complete.

## Updating a system

You can update and restart the following components at any time without stopping a running DigitalEdge system:

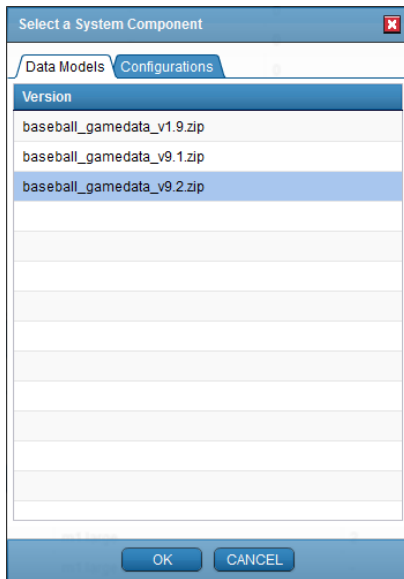
- Data model
- Transport: after you have edited its parameters (not for adding new nodes)
- Data sink: after you have edited its parameters (not for adding new nodes)

When you complete your changes to a data model in the **Data Model Editor**, or to a transport or data sink in **System Builder**, you must update the system in **Management Console** to use the revised component.

1. Edit and save the revised component:
  - In **Data Model Editor**, make sure Save As the revised data model so you can assign the latest minor version number to the data model
  - In **System Builder**, be sure to associate each updated transport or data sink with a **Config Version** number
3. Go to the **Management Console**.
4. Select the system that you are working with. Make sure its status is **OK**.
5. Click the **Update System** icon . The **Select a System Component** dialog box appears.

★ You can update one component at a time in **Management Console**. If you have edited multiple components, you must **Update System** individually for each revised component.

6. Select either the **Data Models** tab to update a data model, or the **Configurations** tab to update a transport or data sink.
7. Click on a **Version** of the data model or system that is ready for updating.



8. Click **OK**. DigitalEdge builds the system and pushes the updated component to the ingest nodes and any data sink that uses ingest.



To restart process groups, [See "Stopping and restarting a process group" on page 32.](#)



To shut down and restart a complete DigitalEdge system, [See "Shutting down and restarting a system" on page 31.](#)

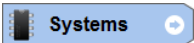

## Deleting a system

If you have a configured system that you are no longer using, you can permanently delete it from DigitalEdge. For example, if you created several test systems before going into production, you may want to delete older versions of a system.



You cannot delete a live system. The system status must be New, Down, or Error before you can permanently delete it.

In the **Management Console**:

1. Click the **Systems** option. 
2. Click the **System** name you want to work with.
3. Click  **Delete System**: to permanently delete a system.

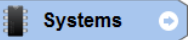

★ When you delete a system, its master configuration file, all previous configuration files, and all data records are erased. If you need to keep the data records, do not delete the system unless you have a backup.

---

## Deleting a system configuration

If you have an older system configuration that you are no longer using because you have revised or updated your system build, you can permanently delete an older configuration file from DigitalEdge. For example, if you created several test configurations before going into production, you may want to delete older system definitions.

In the **Management Console**:

1. Click the **Systems** option. 
2. Click the **System** name you want to work with.
3. Click the **Configurations** tab in the bottom pane.
4. Highlight an older configuration that you want to get rid of.
5. Click  **Delete System Configuration** to permanently delete a system.

★ If you delete the current configuration, DigitalEdge will delete the entire system and all its data records (as if you had deleted the system, not just a configuration file).

---

## Managing security groups and rules

A security group controls incoming communications to a server or process by implementing a set of firewall rules. DigitalEdge security groups are modeled on EC2™ security groups, which restrict communications based on protocol (TCP, UDP, etc.), IP address, and port. DigitalEdge security groups are predefined and automatically assigned to each process group specified in **System Builder** (e.g., JMS, ingest, data sinks, web apps). A security group authorizes work with a process group, provides the ability to open a port for use, and specifies what outside networks can communicate with a process group.

★ You cannot create, edit, or delete a security group. But you can specify security rules associated with a security group.

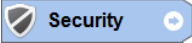
---

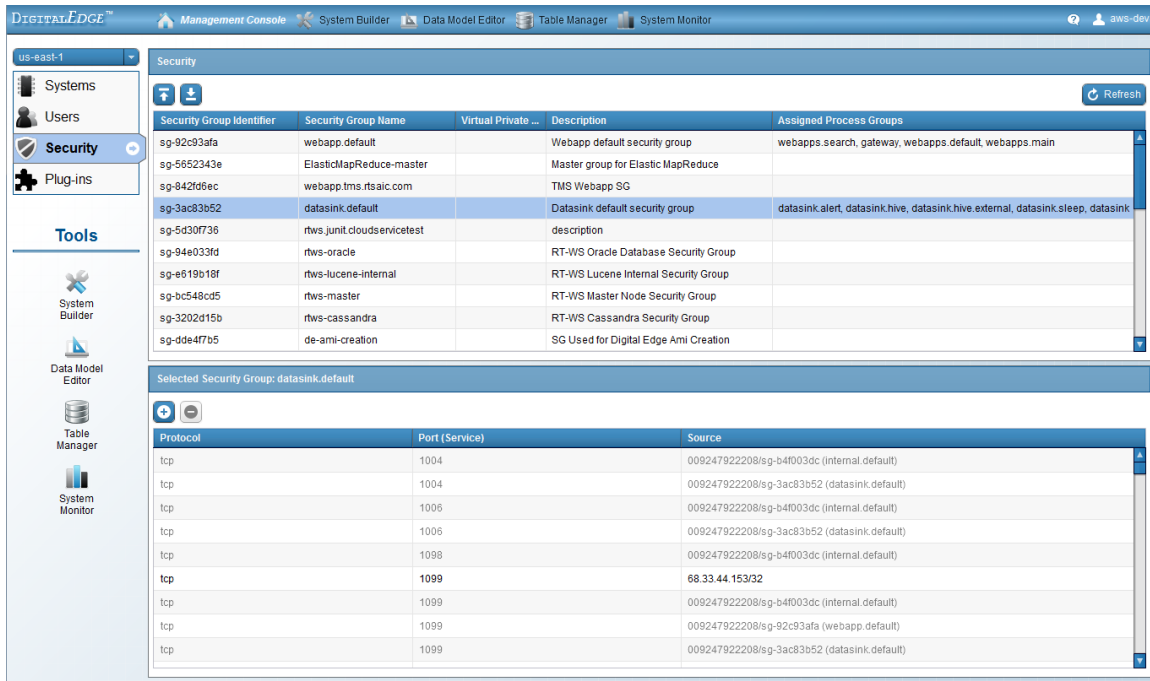
Security group parameters are specified as security rules. A security rule is a permission or a firewall ACCEPT rule. Each rule specifies a communications protocol, a port, and a source (either IP address or another security group that can talk to a process). While you cannot create or edit security groups, you can specify the security rules that are associated with a security group. Use the Management Console's **Security** section to work with security rules. The **Security** section is automatically populated with default security group settings based on the component choices you made in **System Builder**.

Use the **Security** section of the **Management Console** to:

- Open a port for a newly added instance/component (added in **System Builder**).
- Check an instance that is associated with a security group
- Determine who can connect to a process
- Define a security rule for a security group
- Upload or download DigitalEdge certificates, keystores, and truststores for web apps, the Data Transfer Utility, etc.

## Add a security rule to a security group

1. Open the **Management Console**.
2. Click the **Security** option. 
3. From the **Security** list, click the row of a security group you want to work with. The list of security rules appears in the bottom panel.

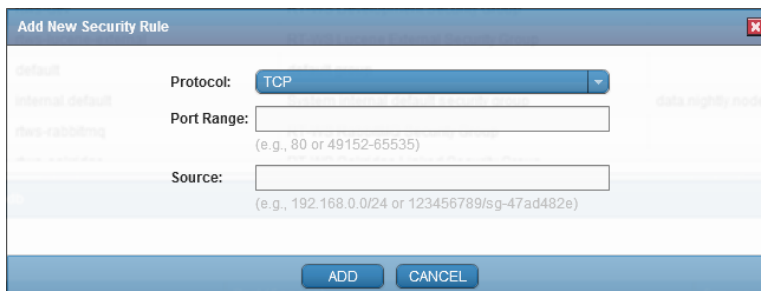


Security Group Identifier	Security Group Name	Virtual Private ...	Description	Assigned Process Groups
sg-92c93afa	webapp.default		Webapp default security group	webapps.search, gateway, webapps.default, webapps.main
sg-5652343e	ElasticMapReduce-master		Master group for Elastic MapReduce	
sg-842fd6ec	webapp.tms.fta.alc.com		TMS Webapp SG	
sg-3ac83b52	datasink.default		Datasink default security group	datasink.alert, datasink.hive, datasink.hive.external, datasink.sleep, datasink...
sg-5d30f736	rtws.junit.cloudservicetest		description	
sg-94e033fd	rtws-oracle		RT-WS Oracle Database Security Group	
sg-e619b18f	rtws-lucene-internal		RT-WS Lucene Internal Security Group	
sg-bc548cd5	rtws-master		RT-WS Master Node Security Group	
sg-3202d15b	rtws-cassandra		RT-WS Cassandra Security Group	
sg-dde47b75	de-ami-creation		SG Used for Digital Edge Ami Creation	

Protocol	Port (Service)	Source
tcp	1004	009247922208/sg-b4f003dc (internal.default)
tcp	1004	009247922208/sg-3ac83b52 (datasink.default)
tcp	1006	009247922208/sg-b4f003dc (internal.default)
tcp	1006	009247922208/sg-3ac83b52 (datasink.default)
tcp	1098	009247922208/sg-b4f003dc (internal.default)
tcp	1099	68.33.44.153/32
tcp	1099	009247922208/sg-b4f003dc (internal.default)
tcp	1099	009247922208/sg-92c93afa (webapp.default)
tcp	1099	009247922208/sg-3ac83b52 (datasink.default)

4. Click the **Add Security Rule**  button. The **Add New Security Rule** dialog box appears.



**Add New Security Rule**

Protocol: **TCP**

Port Range:   
(e.g., 80 or 49152-65535)

Source:   
(e.g., 192.168.0.0/24 or 123456789/sg-47ad482e)

**ADD** **CANCEL**

5. Select a **Protocol** from the drop-down menu (TCP, UDP, or ICMP).
6. Enter a **Port Range** (one port number or a range of ports).
7. Enter a communications **Source**. A source can be an IP address, a range of IP addresses, or another security group that can talk with the assigned process groups.
8. Click **ADD**.

For example, suppose you used **System Builder** to add a data sink that will write to an external database on your corporate system. Select and configure a data sink in the **System Builder**, then locate the data sink in the **Management Console's Security** section. Click on the security group to see the security rules associated with it. You could then **Add** a new security rule to open a port to attach the data sink to the external database.



You cannot edit a security rule. To change a rule, **Delete** it and **Add** it again with revised parameters.


---



Rules which specify standard ports cannot be deleted. These rules are grayed out in the Security Rule List.

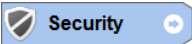


---

### Delete a security rule

1. From the **Security** list, click the row of the security group you want to remove. The list of security rules appears in the bottom panel.
2. Click the rule you want to remove.
3. Click the **Delete**  button.
4. In the **Confirmation** dialog box, click **Yes** to delete the security rule.

### Upload or download security artifacts

DigitalEdge provides the ability to upload or generate certificates (used primarily for web apps), truststores, and keystores (often used with the Data Transfer Utility).

1. Open the **Management Console**.
2. Click the **Security** option. 
3. Click the security group you want to work with.
4. Click the **Upload**  button or the **Download** button .
5. Specify a **Certificate**, **Keystore**, or **Truststore**.
6. Click **Upload....**
  - a. Specify the upload location.
  - b. Browse to the file you want to upload.
  - c. Click **Upload**.

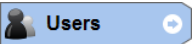

Or click **Download....**


- a. **Open** or **Save** the file.
- b. Click **OK**.

## Managing users

All primary and secondary tenants on an account in DigitalEdge share a user instance, LDAP, and private data store for users. User accounts are created in DigitalEdge for anyone who will be running and using a web application. Note that *all* users are authorized to run *all* end-user web applications; you do not have to associate users with individual web apps. When you create users, DigitalEdge sets up an LDAP system for your system.

### Add a user

1. Open the **Management Console**.
2. Click the **Users** option.  The Users workspace appears.
3. Click **Add User**.  The **Create New User** dialog box appears.




The 'Create New User' dialog box is a light blue window with a title bar. It contains six text input fields with placeholder text: 'Enter the username', 'Enter the description', 'Enter the first name', 'Enter the last name', 'Enter the password', and 'Re-enter the password here'. The fields are arranged vertically. At the bottom of the dialog, there are two buttons: 'ADD' and 'CANCEL'.



4. Create and enter a **Username**.
5. Enter a **Description** of this user.
6. Enter the user's **First Name**.
7. Enter the user's **Last Name**.
8. Enter a security **Password** (there are currently no rules for constructing passwords).
9. Re-enter the password for confirmation.
10. Click **ADD**.

When you need to edit the user list, you can scroll through the list if it is short. If you have a long list of users to maintain, you can jump to an alphabetical starting point in the list. Or, you can search for a user that you wish to view, edit, or delete.


### Narrow down a user list

1. In the **Management Console**, click the **Users** option.  The Users list appears.
2. Click on a **First Letter** option to quickly scroll through the user list to a specific alphabetical point.


### Search for a user

1. In the **Management Console**, click the **Users** option.  The Users list appears.
2. In the upper right corner, enter a string to search on. The string can be all or part of a username, first name, or last name. The string can be embedded anywhere in the user name.
3. Click **Filter**.  A list of matching users is compiled.

### Edit a user

1. From the user list, click the **Edit** button  to the right of the user you want to change. The **Update User** dialog box appears.
2. You can edit any field except the **Username**.
3. Click **UPDATE**.

### Delete a user

1. From the user list, click the **Delete** button  to the right of the user you want to remove.
2. In the **Confirmation** dialog box, click **Yes** to delete the user.

## Managing the component repository

DigitalEdge uses a Master Repository to store all the common and private plug-in components available to your DigitalEdge installations. This repository drives the list of system features available in the Setup Tools (System Builder, Data Model Editor). The Master Repository resides in the TMS environment where only high-level TMS Administrators can access files.

The repository for a primary tenant is sized at 125 MB. This means that:

- You can have a maximum of 7 systems in the starting state at any one time across all secondary tenants in your account.
- DigitalEdge comes with many plug-in components, which occupy 60 MB.

- When you create new plug-ins and scripts, you have a maximum of 65 MB available for repository storage.
- When you upload a new plug-in or script, if the new component will exceed the available storage space, DigitalEdge will warn you and prevent the upload.

DigitalEdge ships with many components that help you build systems. Frequently, however, the standard plug-in components are not sufficient for your site. You may need an additional script or a custom transport for ingesting data. This is where the DigitalEdge *SDK* comes into play. Use the SDK to create:

- Transports
- Data sinks
- Parsers
- Enrichments

Once you create a custom plug-in, you need a way to get it into the Master Repository for safe storage and to access the component for downloading into a DigitalEdge system. The **Plug-ins** feature of the **Management Console** provides this functionality. The functions in the **Plug-ins** section help you upload a custom component to the Master Repository from your hard drive. From there, you can then include the custom component in your DigitalEdge installation with **Data Model Editor** (parsers and enrichments) or **System Builder** (transports and data sinks).



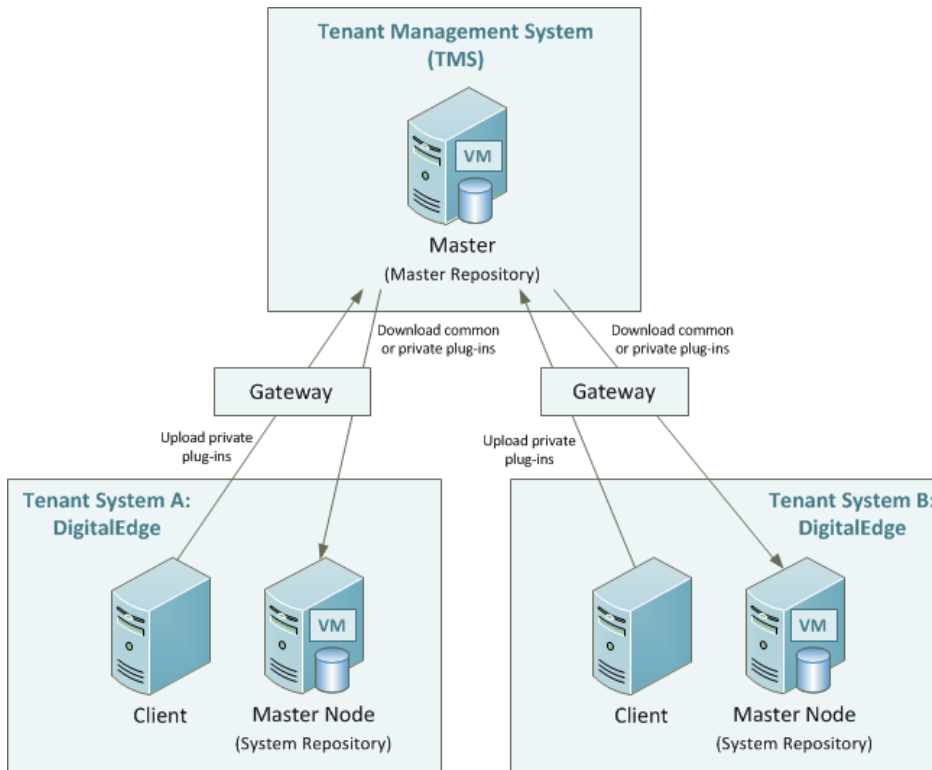
To integrate a data sink with DigitalEdge, contact a DigitalEdge Support Engineer to assign it to a process group.

---

You usually do not have to manually download these plug-ins to your tenant System Repository; downloading is automatically done at system start-up and re-start. Plug-ins are triggered for system download as you specify components to use from the lists in **System Builder** and **Data Model Editor**. Downloadable plug-ins include:

- Transports
- Parsers
- Enrichments
- Scripts





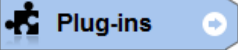

You can upload the following types of plug-ins to the Master Repository:

- **Software:** Custom created plug-in components such as transports, enrichments, and parsers
- **Scripts:** Scripts are used in two places in DigitalEdge: to drive a scripting data sink in **System Builder**, or to specify a non-standard data source translation in **Data Model Editor**


### Getting custom components into the repository

After you create a custom plug-in with the DigitalEdge *SDK*, you need to upload the component into the TMS Master Repository from your hard drive. In the **Management Console**, the **Plug-ins** section provides this functionality. Once a component is in the Master Repository, you can select and download it in **Data Model Editor** or **System Builder** just as you would a common component (that is, a component provided with DigitalEdge).

#### Upload a custom (private) plug-in to the Master Repository

1. Access the **Management Console**.
2. Select the **Plug-ins** section. 
3. On the **Software** tab, double-click  **Private Software**.

---


★ You can double-click on **Common**  to see a list of components that are in the Master Repository, that are provided with DigitalEdge, and that are downloaded when you **Start** a system. But you cannot upload, delete, or manipulate files and folders in the **Common** area. If the component list is long, you can narrow down the file list with a filter string in the **Start Typing to Filter** box.

---

- Next, you must specify the Master Repository folder where your plug-in(s) will reside. Either choose an existing folder in the Master Repository or create a new folder.

Double-click on an existing folder in the Master Repository as the destination.

or,

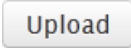
Click **New Folder**  to create a new destination folder in the Master Repository.

- Click **Upload** . The **Upload Software** dialog box opens.
- Click **Browse** to locate and select the private component from your hard drive.

---

★ You can **Upload** a transport, parser, or enrichment plug-in of file type .JAR or .ZIP. To upload a data sink, contact Leidos.

---

- Click **Upload** . If the file is not in the appropriate JAR file format, an error message will appear.

---

★ Each private component must be packaged in a jar file, should contain a pom.xml file, and should not contain nested jars, as specified in the standard at <http://maven.apache.org/>.

---



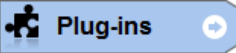

---

★ If the new component will exceed the available storage space, DigitalEdge will warn you and prevent the upload.


---

- The plug-in now appears in the list of available components on the **Software > Private Software** tab. It is also now selectable in **System Builder**.

### *Upload a private script file to the Master Repository*

- Access the **Management Console**.
- Select the **Plug-ins** section. 
- On the **Scripts** tab, double-click  **Private Scripts**.

---


★ You can double-click on **Common**  to see a list of components that are in the Master Repository and provided with DigitalEdge, but you cannot upload, delete, or manipulate files and folders in the **Common** area.


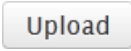
---

- Next, you must specify the Master Repository folder where your script(s) will reside. Either choose an existing folder in the Master Repository or create a new folder.

Double-click on an existing folder in the Master Repository as the destination.

or,

Click **New Folder**  to create a new destination folder in the Master Repository.

- Click **Upload** . The **Upload Scripts** dialog box opens.
- Click **Browse** to locate and select the private script from your hard drive. You can upload .SH, .PY, PL, or .GROOVY files.
- Click **Upload** .





If the new component will exceed the available storage space, DigitalEdge will warn you and prevent the upload.

---

- The plug-in now appears in the list of available components on the **Scripts > Private Scripts** tab. It is also now selectable in **System Builder**, **Data Model Editor**, or the **Periodic Tasks** section of the Management Console.

### Create a new folder in the Master Repository

- Access the **Management Console**.
- Select the **Plug-ins** section. 
- Select either the **Scripts** or **Software** tab.
- Double-click the **Private** option.
- Click **New Folder**  or double-click on an existing folder first to create a lower level folder.
- In the **Make New Folder(s)** dialog box, enter the **New Folder(s)** name, separating nested folder names with a forward slash /, then click **OK**.
- You can now upload files to the new folder(s).

### Deleting a custom component

If you have a private plug-in stored in the Master Repository that you no longer need, you can permanently delete it. Once deleted from the Master Repository, a plug-in will not appear in the master list of available components in **System Builder**.

### Delete a private file or folder from the Master Repository


1. Access the **Management Console**.

2. Select the **Plug-ins** section. 

3. Select either the **Scripts** or **Software** tab.



4. Double-click the **Private** option.

---

★ You cannot delete a component from the **Common**  area.

---

5. Double-click on the source folder in the Master Repository if there are multiple folders to choose from.

6. Click on the file  or folder  that you want to delete. (Ctrl-click to deselect an item.)

---

★ When you delete a folder, all its contents will also be deleted (subfolders and files).

---

7. Click the **Delete** icon .

8. The **Confirm Delete** dialog box appears. Click **Delete** to confirm, or click **Cancel** to keep the plug-in.

---

★ Delete cannot be undone. Be sure you want to delete the file from the Master Repository. Also note that this function does not delete any components in your DigitalEdge system. If you restart your system (in the **Management Console > System**), the deleted plug-in will be removed from your DigitalEdge system.

---

### Creating and scheduling jobs

A periodic task is a job that you script and run on a regularly scheduled basis. For example, you may want to create and schedule jobs to:

- Tar up log files to create and store an archive of DigitalEdge logs
- Archive any DigitalEdge files
- Automatically stop an inactive system from running at night and restart it in the early morning for users, to reduce cloud costs
- Run a map reduce job nightly to generate analytical reports
- Connect to a transactional database nightly to update DigitalEdge data sinks on a scheduled basis; this may be useful if a database is not available in real time as a data source for DigitalEdge (e.g., your organization has a policy restricting access during business hours)

Tasks should be written as shell scripts and stored in the root directory of your script repository.


---

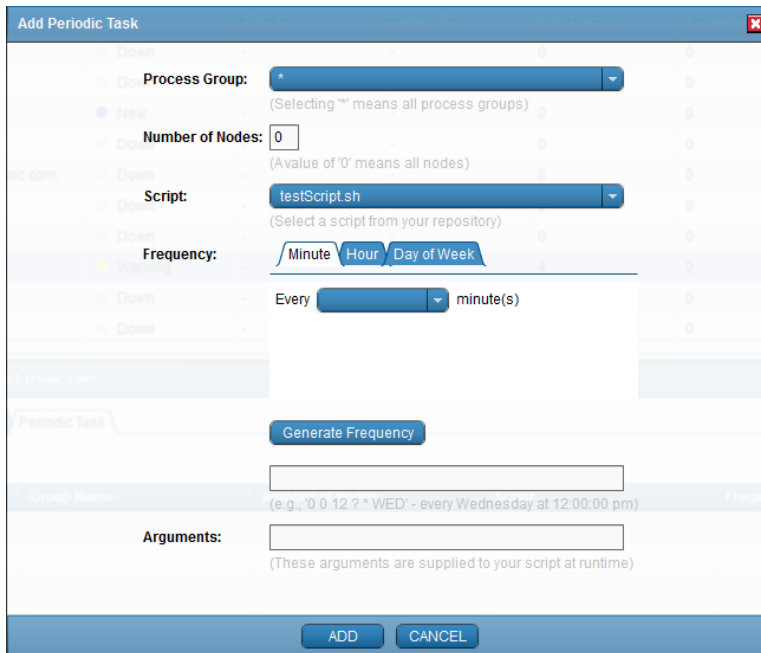
★ You can use any scripting language that can be run on a command line (e.g., Bash, Perl, etc.).

---

Use the **Management Console > Systems > Periodic Tasks** tab to create and schedule jobs.

### Create a scheduled job

1. Write a shell script for your task.
2. Save the script in the root directory of your private scripts repository (**Management Console > Plug-ins > Scripts > Private Scripts**). [See "Getting custom components into the repository" on page 41](#) for details.
3. In the **Systems** section of the **Management Console**, highlight the system you want to work with.
4. Click the **Periodic Tasks** in the bottom panel. Click the **Add** icon. 
5. Create and schedule the task:



- **Process Group:** Select the process group you want to work with, or \* to select them all.
- **Number of Nodes:** Indicate how many sequential nodes to work with; 0 = all, 2 = the first and second nodes, etc.
- **Script:** Select the shell script you wrote and stored in the root directory of your private scripts repository.
- **Frequency:** Define the schedule for running your script by selecting one tab to work with:
  - **Minute:** Select the interval in which to run the script, such as every 30 minutes. Scripts are run on a multiple of the minutes selected. For example, select 10 to run a job starting at the next interval of 10 minutes. If it is currently 1:14, the script will first run at 1:20, and then repeat every 10 minutes after that: 1:30, 1:40, etc.

- **Hour:** Select the hourly interval for running the script, such as every 8 hours. Scripts are run on a multiple of the hours selected. For example, select 2 to run a job every 2 hours (starting at the next hour divisible by 2). If it is currently 5:10 PM, the script will first run at 6:00 PM (the next even-numbered hour), and then repeat every 2 hours after that: 6:00 PM, 8:00 PM, etc. (One special case: If you select 13, the job will run at midnight and 1:00 PM consistently.)
- **Day of Week:** Click a day for running the script and select the **Start Time** for kicking off the job. This script will run once a day.

Click **Generate Frequency** to see the defined schedule in the text box. You can edit this text, or enter your own definition for a more complex frequency (e.g., run a job every 12 hours only on weekends).

- **Arguments:** Enter any run-time arguments that your script requires.

6. Click **ADD**. The new task and details are listed on the **Period Task** tab.



You cannot edit a periodic task; to change a task, delete it and re-create it.

---



To determine if a periodic task was scheduled and actually ran, access the `/logs/default.log` file.

---

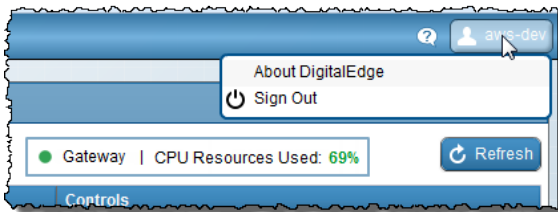
## Delete a scheduled job

1. In the **Systems** section of the **Management Console**, highlight a job in the **Periodic Tasks** list at the bottom of the screen.
2. Click the **Delete**  button.

## Logging out

Use this procedure to log out of DigitalEdge.

1. Go to the **Management Console**.
2. Click the user icon in the upper right corner and select **Sign Out**.



Use the same procedure to log out of any Setup or Runtime UI tool.

---



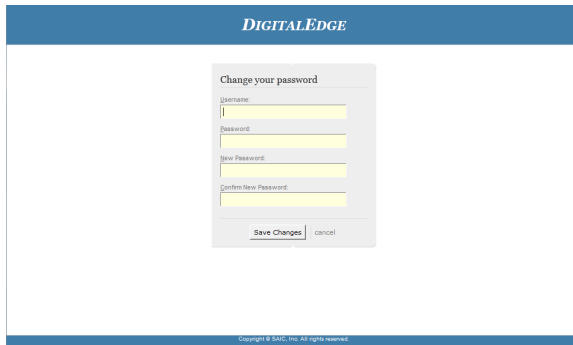
When you **Sign Out** of one tool, all open tools are automatically signed out.

---

## Changing passwords

Use this procedure to periodically change your DigitalEdge password.

1. In a web browser, go to [https://<system\\_domain\\_name>/tenantconsole](https://<system_domain_name>/tenantconsole)
2. Click **Change Password**.



The screenshot shows a web browser window with the DigitalEdge logo at the top. In the center, there is a 'Change your password' form. The form contains four input fields: 'Username', 'Password', 'New Password', and 'Confirm New Password'. Below these fields are two buttons: 'Save Changes' and 'cancel'. At the bottom of the browser window, a small copyright notice is visible: 'Copyright © 2012, Inc. All rights reserved.'

3. Enter your **Username**, current **Password**, and **New Password**.
4. Enter the new password again to **Confirm New Password**.
5. Click **Save Changes**.







## Chapter 6: System Monitoring

A number of points exist throughout DigitalEdge to monitor data flow and normal operations. These access points include:

- **Management Console:** To get a quick snapshot of a running system, including what process groups are running, the number of instances and types, and the number of volumes and sizes; [See "Viewing system status" on page 49](#)
- **Log files:** Each DigitalEdge node runs a log file that records startup processes and problems; [See "Checking log files" on page 52](#)
- **System Monitor:** A dynamic console tool for monitoring system health and potential problems in real time, visually depicts system activity and resource scaling; [See "Checking system metrics" on page 55](#)

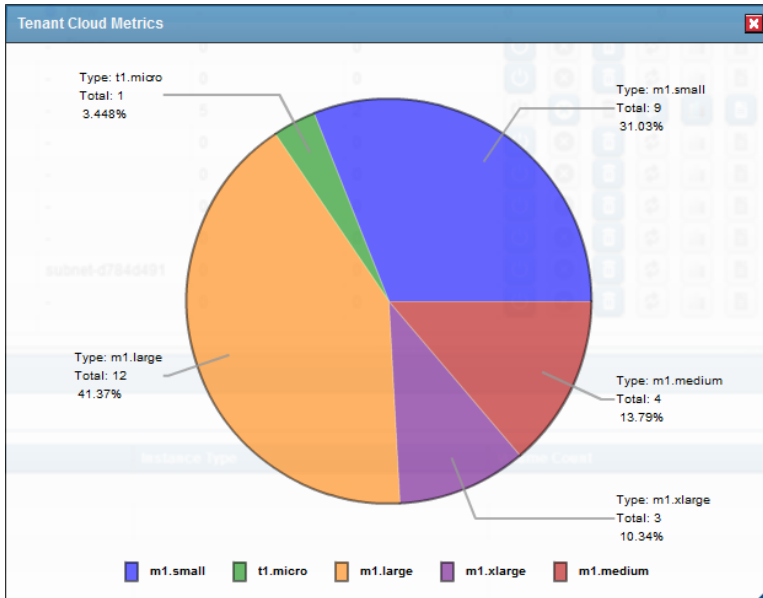
### Viewing system status

You can get a quick snapshot of a running system in the Management Console. This is not a dynamic view like you see in the System Monitor. This snapshot tells you what process groups are running, the number of instances and types, and the number of volumes and sizes. The view is static, not auto-refreshed. Click Refresh  to view the latest status.

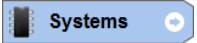
1. Open the **Management Console**.
2. The status of the **Gateway** node  Gateway | CPU Resources Used: 31% appears in the top right (green = OK, yellow = Warning, red = Error).
  - a. Click **Gateway** to view its name and IP addresses for troubleshooting.
  - b. The **Gateway** icon also includes the current percentage of available **CPU Resources Used** for the node:






% CPU Resources Used	State
0 - 74%	Green = OK
75 - 89%	Orange = Caution: Consider adding resources
90 - 100%	Red = Warning: Increase resources now
ERR	The Gateway status refreshes every 5 seconds; an Error status may resolve itself. If the Gateway stays in ERR mode for several minutes, call DigitalEdge Technical Support.
???	When the Gateway first starts up, its initial status is listed as ???. This status should resolve to a normal state within seconds.



- c. Click **CPU Resources Used** to view a pie chart that details resource utilization for the virtual machines in your tenant cloud environment:



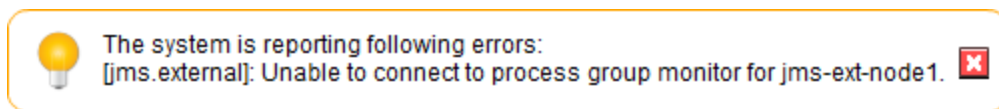
This chart summarizes the total resources used for each instance in your tenant account. To increase available resources, first shut down all unnecessary DigitalEdge systems running in Management Console (test systems, prototypes, outdated systems, etc.). If available resources are still less than 25%, you should consider sizing up. On an AWS cloud, increase your instance limits to allow for the creation of more instances. On a Eucalyptus-based system, add more hardware. In either case, call DigitalEdge Technical Support for assistance.

3. Click the **Systems** option . A list of configured systems displays the status of each system. As a system is created and then started up, the system status should cycle through **New** > (Starting) > **OK**.

Status		Description
	Green	<b>OK:</b> All nodes have started, everything is functioning normally, the database is valid.
	Blue	<b>New:</b> A reminder that you created and built a new system, but haven't run/started the system yet. Click the <b>Start</b> icon to run the system. 
	Light gray	<b>Down:</b> The master node matches data in the cloud, but the cloud has stopped working.
	Yellow	<b>Warning:</b> The system's master node could not find information about process groups; DigitalEdge cannot determine if the system is running. The system will keep trying to start. Click on

Status		Description
		the status to see more detailed information in the bottom panel.
	Red	<b>Error:</b> TMS cannot locate the system that is stored in the database. Click on the status to see more detailed information in the bottom panel. You can delete and recreate the system or contact Leidos for assistance.
	Dark gray	<b>Unknown:</b> The system is transitioning between statuses.

- a. Click the **System** name you want to check on. If a node is down, a warning will appear in the message box, such as:





- b. The system status report appears in the bottom panel, listing system resources which are currently running in the cloud. Specific information about each node appears on the **Process Groups** tab:

Column	Explanation
<b>Group Name</b>	A process group represents a choice of components or applications that you made in System Builder (such as a web app or a data sink) that is currently configured in your system. A process group represents a collection of instances in each category (e.g., all instances of a JMS internal queue).
<b># Instances</b>	How many instances of the process are currently running
<b>Instance Type</b>	The size of each node, as defined by Amazon's node instance types (small, medium, large, Xlarge)
<b>Volume Count</b>	How many volumes are currently consumed by a process group
<b>Volume Size</b>	How large each volume is, in gigabytes
<b>Auto Scaling?</b>	Whether or not a process is self-scaling

- c. Double-click on a process **Group Name** to see a list of instances and IP addresses within that group. Note that this information is read-only:

System: phoenix.rtsaic.com   Process Group: hadoop.hbase.datanode.regionserver				
Copy				
Name	Instance ID	Persistent IP	Public IP	Private IP
dtanrgs1.phoenix.rtsaic.com	i-157cba35	-	50.17.100.113	10.216.128.12
dtanrgs2.phoenix.rtsaic.com	i-a57abc85	-	54.197.95.121	10.153.133.57

- d. The **Configurations** tab lists all of the system's configuration versions. You can **Delete**  any configuration file that is not currently in use; you cannot delete the active configuration file.
- e. Click **Refresh**  to update the status report.




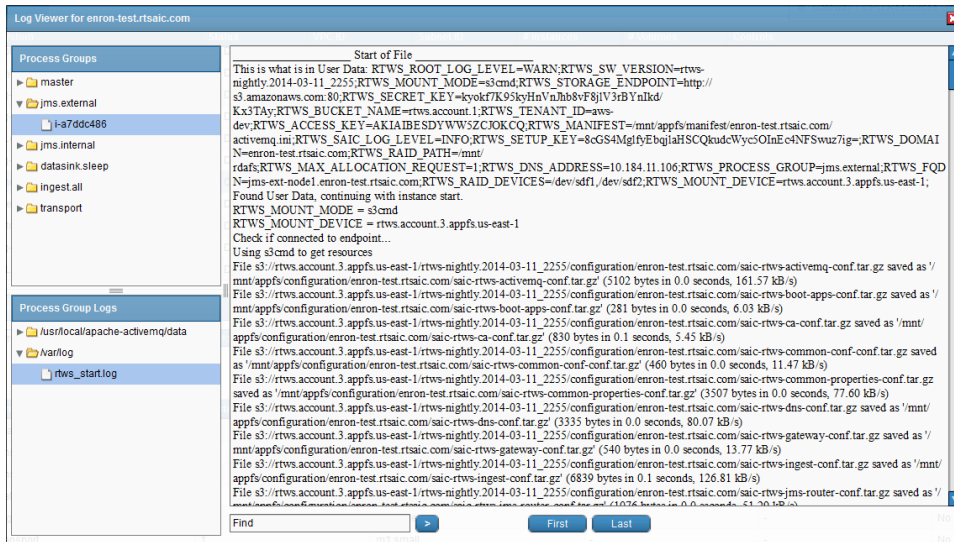
To edit system parameters, [See "Process group parameters" on page 82](#)

## Checking log files

DigitalEdge offers a logging and audit framework which includes the generation, review, protection, and retention of logs across the virtualization, operating system, application, and data layers. The framework also provides customizable logging levels.

Each instance has its own set of log files. You can access all the DigitalEdge log files from one convenient place in the Management Console.

- In the **Management Console**, click the **View Logs** icon  on the line representing the system you need to work with. The **Log Viewer** appears, with information in three panels:
  - Process Groups:** Lists the process groups and nodes that have started up and have log files
  - Process Group Logs:** Lists the folders and log file names on each node
  - Log Content:** Provides a dump of a selected log file



2. Expand any **Process Group**, such as:

- master
- jms.external
- jms.internal
- datasink.lucene
- webapps.main
- ingest.all
- transport

to see a list of nodes assigned to each process group, such as:

- ingest.all
  - i-bcabd2c4

3. Click on a node to view the **Process Group Logs** on that node. The hierarchical list includes directory paths and log file names, such as:

- /usr/local/rtws/ingest/logs
  - console.log
  - ingest.log
  - metrics.log

4. Click on a log file name to view its content. Lengthy files are paged in the Log Viewer; use the vertical scroll bar or click **First** and **Last** to move through a file. You can also search for a string and find each next occurrence  **>**. Note that search strings are case sensitive.



The master node controls the initialization process for all other DigitalEdge nodes. The `master.log` file contains much of the information associated with system initialization.

For example, the following commonly accessed logs are useful when troubleshooting:

Troubleshooting Area	Log Files
Master node and system initiation	usr/local/rtws/master/master.log
Startup problems for individual nodes	var/log/rtws_start.log
Ingest issues	usr/local/rtws/ingest/logs
Data sink tracking	<div>usr/local/rtws/ingest/logs<ul style="list-style-type: none"><li>• console.log - data recorded before the data sink runs</li><li>• ingest.log - processing details while the data sink runs</li><li>• metrics.log - statistics reported while the data sink is running</li></ul></div> <div>var/log/hive - data sink installation information, exposing what is running on a specific VM</div>
Webapps	usr/local/jetty/logs



You can also trace the status of unprocessed records in the Dead Letter Queue (DLQ). [See "Checking unprocessed records" on page 55.](#)

## Checking unprocessed records

When a record fails to be parsed or processed, it goes to the Dead Letter Queue (DLQ). You can view the DLQ to determine the problem(s) so you can edit and resubmit failed records.

1. Access the **Data Transfer Utility**:

```
https://default.<domain>/tenantconsole/docs/dtu/
```

2. Run the DTU:

```
Java -jar DataTransferUtility.jar
```

3. Click the **DLQ** tab (or select **Window > DLQ**). You can view the unprocessed records in the DLQ.
4. To reprocess records, you must edit and resubmit them. You can do this by copying records from the DLQ to another file, editing them, and resubmitting them to DigitalEdge.



## Checking system metrics

The System Monitor visually depicts system activity and resource scaling. You can monitor processes as they auto-scale up and down and as resource utilization changes.

The System Monitor is a dynamic console tool. The graphs change on-screen, reflecting the most current system status. You can monitor system health and potential problems in real time, including such items as:

- Ingest data flow rate
- Record processing backlog
- Alerting engine throughout
- Resource consumption and auto-scaling
- Data storage usage
- User application activity

## System Monitor access


1. Access the **Management Console**.
2. Go to the **Systems** page.
3. Click on the **Monitor System** icon  on the row representing the **System** you wish to monitor. Or, click the **System Monitor Tool**.  The **System Monitor** appears on a new browser page.

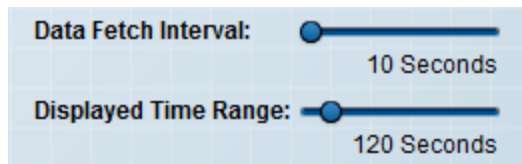
The System Monitor presents a dynamic picture of several key performance factors. There are two groups of graphs:


- Overview graphs
- Detail graphs

## Controls

Graph controls at the top and bottom of the screen change the data views:

- **Monitoring:** Click **Monitoring** to select the domain name of a different system to monitor
- **Show Settings:** At the top right of the screen, click the **Show Settings** icon  to view settings that control the look and performance of all graphs:

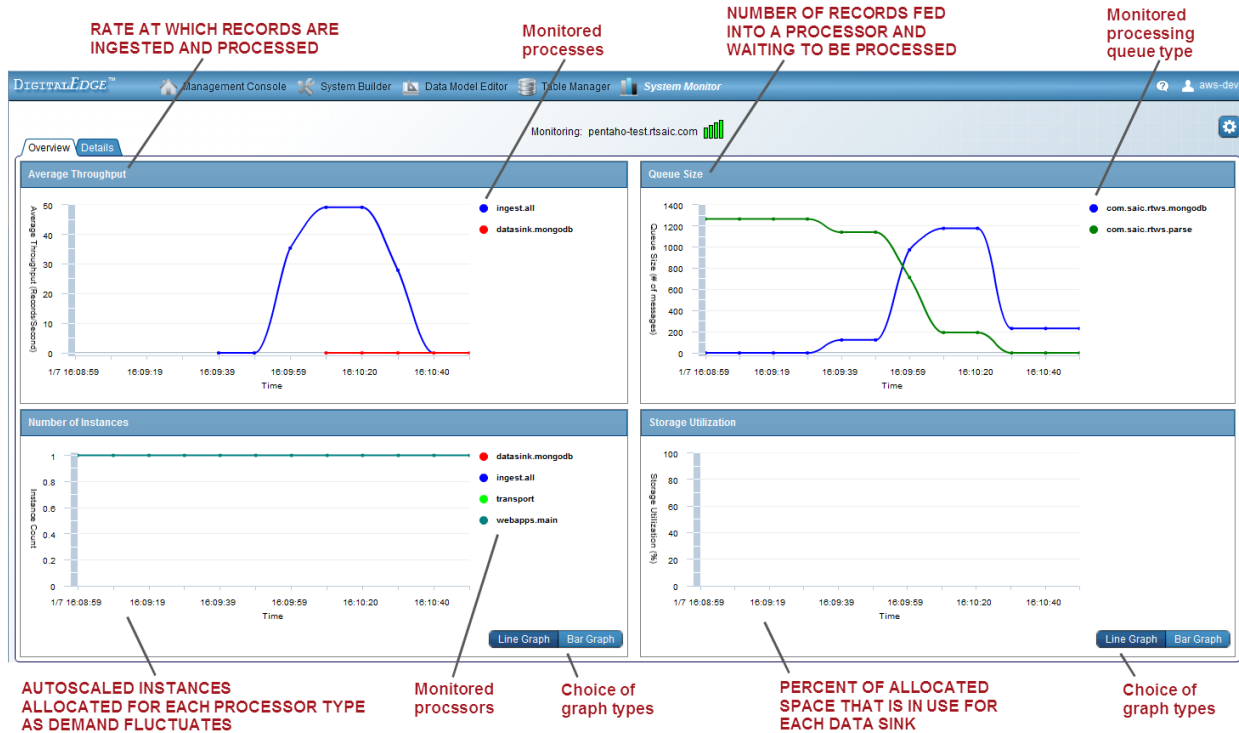


- **Data Fetch Interval:** Use this slider to set the number of seconds between refreshes and to define the intervals listed on the X axes
  - **Displayed Time Range:** Use this slider to set the maximum number of seconds depicted on the scales of the X axes
- **Next Data Fetch:** At the bottom of the screen, the refresh counter displays the number of seconds left before the screen is refreshed.  4 Seconds

## Overview graphs

Click the **Overview** tab to view the basic system graphs simultaneously:

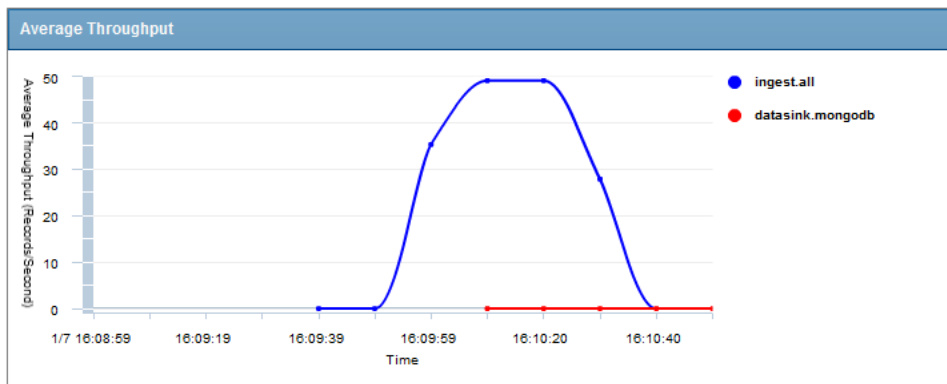




These graphs summarize metrics across all process groups:

- **Average Throughput:** The rate at which ingested records are fed into the system
- **Queue Size:** The number of records that were fed into the system but which are not yet processed
- **Number of Instances:** The number of instances that are automatically allocated and consumed for each process
- **Storage Utilization:** The percentage of allocated space that is currently in use for each data sink which is an EBS (Amazon Elastic Block Storage) volume database

### Monitor data flow: Average Throughput graph



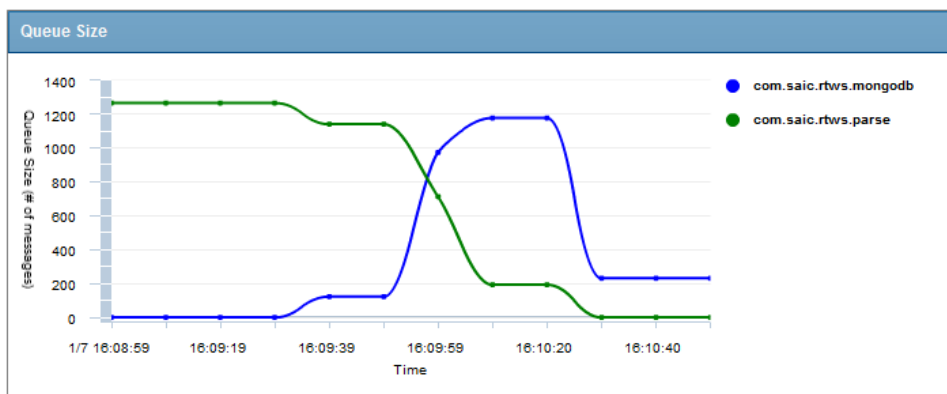
The **Average Throughput** graph illustrates the rate at which records are ingested and processed in the system (records per second). Throughput time is clocked from the second a record enters the

system to the time it is ready for querying (including transport from a data source into the JMS external queue, ingest processing, normalization, enrichment, indexing, and storage). Each line represents records from one process. The legend lists which color line represents each processor.

Use the **Average Throughput** graph to:


- Check how data flow is affecting overall system performance
- Make sure throughput is not sitting at zero, indicating an idle or problematic system
- Check for expected throughput fluctuations based on historical knowledge of peaks and valleys in data feed activity
- Determine how long (on average) it takes to process a record

### Monitor queue sizes: *Queue Size graph*



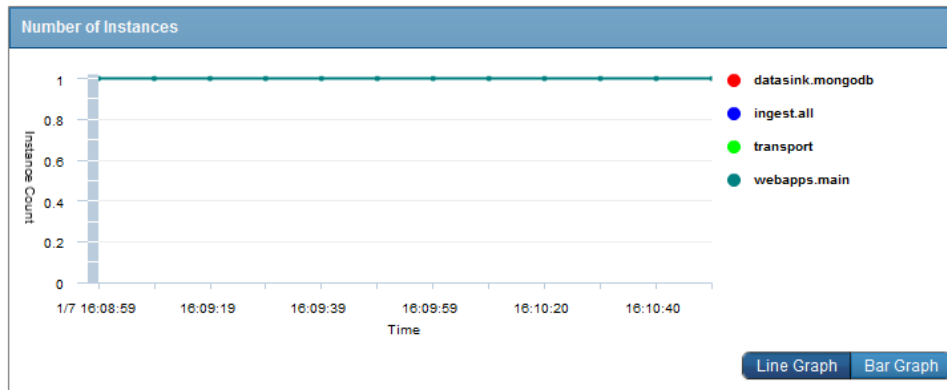
The **Queue Size** graph illustrates how many messages have been fed into a specific processor but which have not yet been processed. This is your processing backlog, which obviously should be as low as possible. This chart monitors different processing queues (identified in the legend on the right), such as:

- **DLQ:** The dead letter queue, where records that could not be parsed are saved
- **Filter:** The alerting engine filters records for potential alerts and notifications
- **Index:** Lucene indexed records in preparation for searching
- **Parse:** The processing pipeline parses, normalizes, and enriches records

Use the scroll bar  at the bottom of the graph to scroll through the timeline on the x axis. Note that the scroll bar only appears if the timeline doesn't fit into the width of the graph/browser window.

Use the **Queue Size** graph to:

- Make sure the backlog queues are as low as possible
- Determine if processing (**Average Throughput**) is keeping up with queue sizes

**Monitor resource scaling: Number of Instances graph**

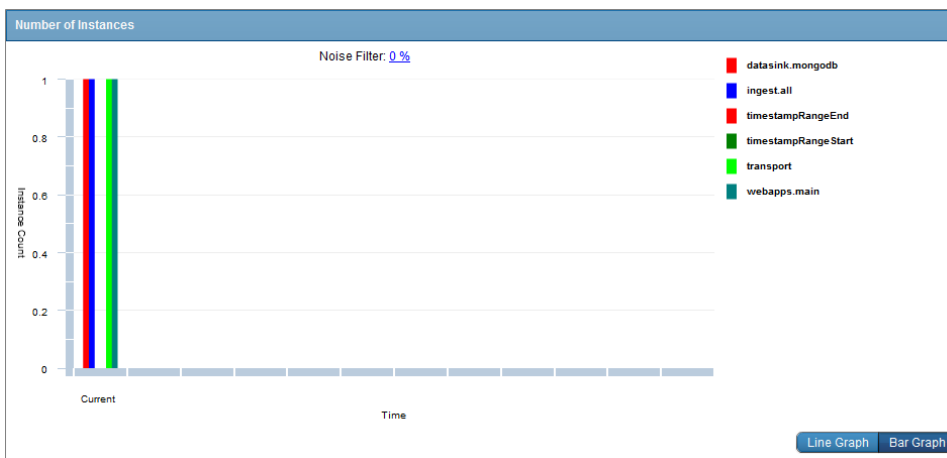
The **Number of Instances** graph depicts how many instances of each process type are allocated at any given time. The graph changes as processes scale up during peak activity and scale down as demand decreases. Each process is represented by a separate color line on the graph; the legend on the right is keyed to specific processors. The **ingest.all** processor is probably the most important process to monitor on this chart; the graph tells you if ingest auto-scaling is keeping up with throughput.

Use the **Number of Instances** chart to:

- Monitor fluctuations in resource consumption
- Decide if ingest processing is allocated sufficient resources
- Determine if indexing (the Lucene processor) is keeping up with ingest
- Verify that user activity (searching via webapps) is not affecting system performance
- Determine if a processor is down

★ Colors are assigned to specific services and synced across all charts.

You can toggle this graph between a **Line Graph** and a **Bar Graph** view:



In the **Bar Graph** view, you can set the **Noise Filter** percentage to specify how frequently the graph should be refreshed:

- 0% = Show all changes to allocated instances (i.e., the noise filter is off); this is the default setting
- 100% = Don't show any changes to allocated instances (a static graph)
- n% = Show increases or decreases in the number of allocated instances only when the delta reaches n% of the total number of instances; you will not see a change in the graph every time the number of instances increases or decreases

### ***Monitor storage usage: Storage Utilization graph***

The **Storage Utilization** chart depicts the percentage of allocated space that is currently in use for each data sink that is an EBS (Amazon Elastic Block Storage) volume database (e.g., Lucene, Hadoop). Each data sink is represented by a separate color line on the graph; the legend on the right is keyed to specific data sinks. This graph is populated when you implement a search app.

Most data sinks are auto-scaling and dynamically expand when a usage threshold is reached. But some data sinks are not auto-scaling and must be monitored for space utilization; in that case, you must take action to allocate more storage when a critical threshold is reached.

Use the **Storage Utilization** graph to:

- Determine if storage is reaching a critical point

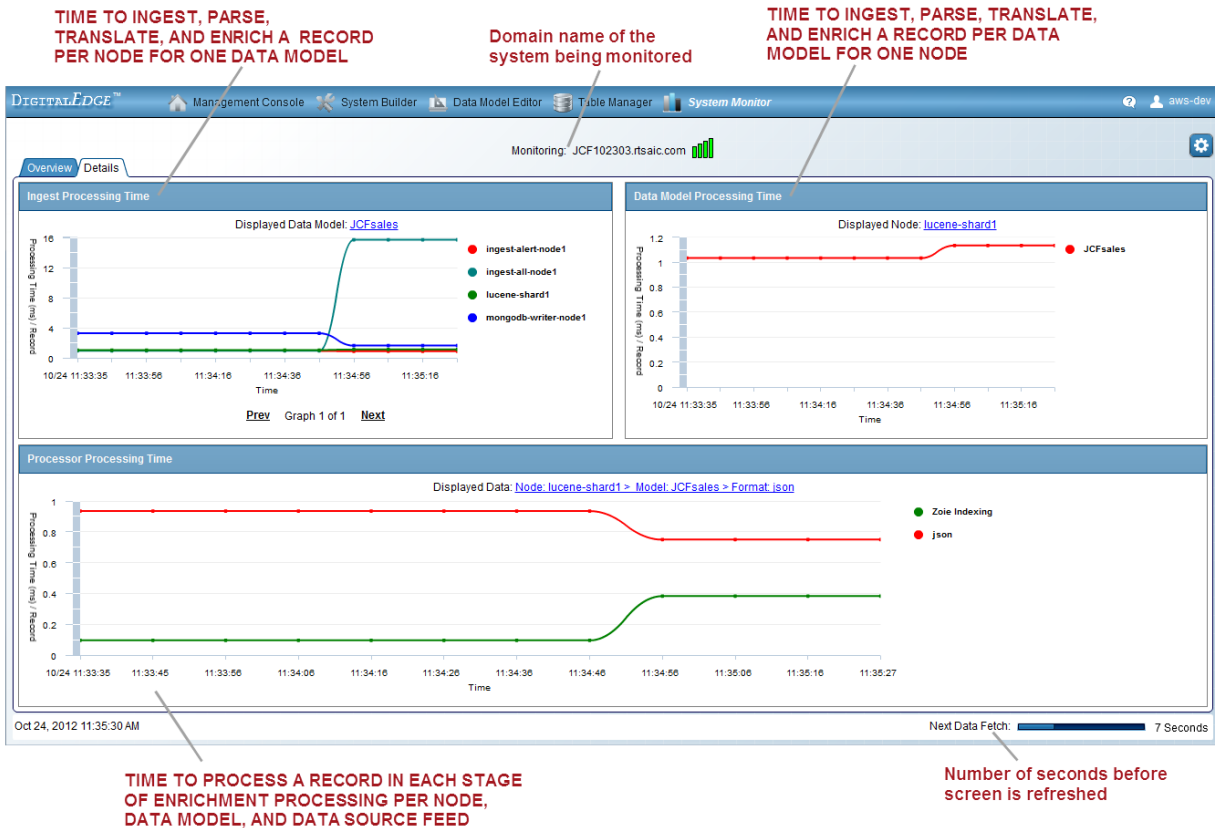
You can toggle this graph between a **Line Graph** and a **Bar Graph** view.

In the **Bar Graph** view, you can set the **Noise Filter** percentage to specify how frequently the graph should be refreshed:

- 0% = Show all changes to storage utilization (i.e., the noise filter is off); this is the default setting
- 100% = Don't show any changes to storage usage (a static graph)
- n% = Show changes in storage utilization only when the delta reaches n% of the total amount of currently utilized storage; you will not see a change in the graph every time storage use increases or decreases

### **Detail graphs**

Click the **Details** tab to access and drill down into the processing time graphs. These graphs help diagnose problems with slow throughput rates:



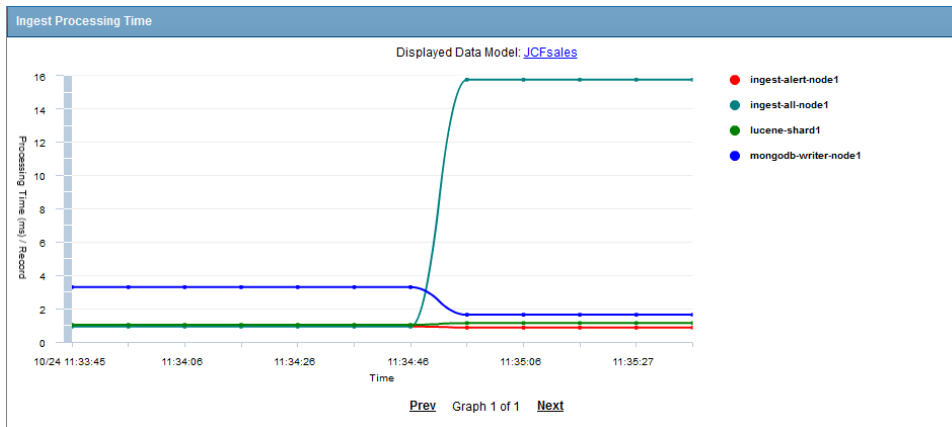
These drill-down graphs break out metrics by a specific processor type; filtered by data model, node, and data source:

- **Ingest Processing Time:** The time it takes (in milliseconds) to ingest, parse, translate, and enrich a record per node for a specified data model
- **Data Model Processing Time:** The time it takes (in milliseconds) to ingest, parse, translate, and enrich a record per data model for a specified node
- **Processor Processing Time:** The time it takes to process a record in each stage of enrichment processing per node, data model, and data source feed



If the Details graphs do not display any data, click the **Monitoring** control and click **Refresh** several times.

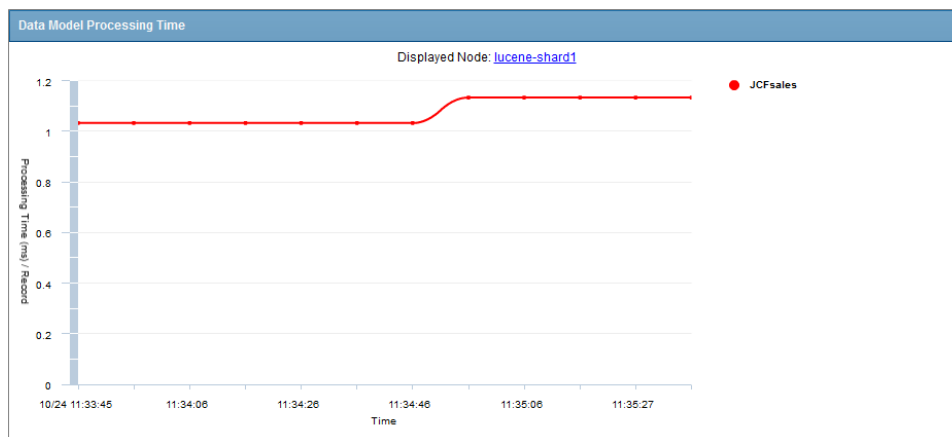
## Monitor ingest per node: Ingest Processing Time



This graph depicts the time it takes to ingest a record, in milliseconds. Processing time includes ingest, parse, translate, and enrichment stages. Times are reported per node (one line per node, keyed to colors in the legend), for a selected data model. Click the **Displayed Data Model** graph option to select one data model to graph. Scroll through the graph with **Prev** and **Next** to compare ingest processing times across nodes.

This graph also charts processing time by data sink.

## Monitor ingest by data model: Data Model Processing Time



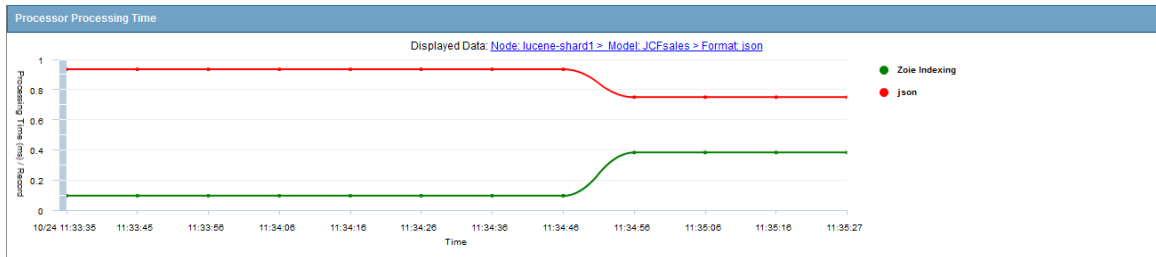
This graph displays the time it takes to ingest a data model record, in milliseconds. Processing time includes ingest, parse, translate, and enrichment stages. Click the **Displayed Node** graph option to view statistics from one specific node. Each line represents records from one data model, keyed to colors in the legend.

This graph also charts processing time by data sink.

For example, use the **Ingest Processing Time** and **Data Model Processing Time** graphs to diagnose problems such as:

- If one line on the **Ingest Processing Time** graph is running slow, drill down to that particular node/data model line in the **Data Model Processing Time** graph for comparison. Ingest may be slow if you have a large dimension table and a cache setting that is too small to efficiently process the data. You may have to configure a larger cache or restructure the dimension table to improve processing times.

### Monitor processing stages: Processor Processing Time



This graph displays the time it takes to process a record in each stage of processing. Each color line (identified in the legend on the right) represents a different processing stage (input, parsing and translating, enrichment, and data sink processing). Click the **Displayed Data** option to drill down through the data by: node, data model, and data source feed.

### Logging out of the System Monitor

When you are done with a System Monitor session, you can leave the console running or log out.

To log out of the System Monitor, click your user icon/name in the upper right corner. Select **Sign Out**.





## Chapter 7: Fine-tuning the DigitalEdge Configuration

Once you have a basic DigitalEdge system configuration and running, you will most likely return to the UI tools to refine the data model, to tweak system parameters, and to add components. Advanced fine-tuning should only be undertaken after you have some experience with a running system to observe status and performance issues.

While the **Management Console** is the most frequently accessed UI tool for checking system status and managing system activity, the **System Builder** is the most frequently used UI tool for editing system parameters and adding more components to a running system. System adjustments include such tasks as:

- **Add a new user application** for indexing data, searching processed data, or specifying alerting applications; [See "Adding a new user application" on page 65](#)
- **Configure an additional data sink** to add storage, alerts, or indexing; [See "Configuring an additional data sink" on page 65](#)
- **Send data to an external application** to use DigitalEdge processed data in another application in your shop; [See "Sending data to an external application" on page 80](#)
- **Resize a process or server** to improve auto-scaling performance; [See "Resizing a process or server" on page 82](#)
- **Fine-tune auto-scaling** for a process; [See "Fine-tuning auto-scaling" on page 81](#)
- **Open a port for a new instance** or component; [See "Opening a port for a new component" on page 84](#)



Once you have configured a system, you cannot change its domain name.




Once you build a system, you can always resize it to a larger size, but you cannot downsize a system once it is configured.

---


### Adding a new user application

You can add a new user app at any time, to index data, for full text searching, for defining alerting rules, etc.

1. Open **System Builder**.
2. Go to the **Overview** tab.
3. Use the **Webapps/REST APIs** section.
4. Click **Add**.
5. Choose an app from the **Select a Webapp/REST API** dialog box and click **OK**.
6. Click **Build**. 

### Configuring an additional data sink

You can add data sinks to your system at any time, for storage, indexing, or alerting.

1. Open **System Builder**.
2. Go to the **Overview** tab.
3. Use the **Data Sinks** section.
4. Click **Add**.
5. Choose a data store from the **Select a Data Sink** dialog box and click **OK**.
6. Edit the default values in the **Set Datasink Parameters** dialog box or click the **Data Models** tab to select a data model to associate with this data sink. Click **OK**.
7. Click **Build**. 

## Data sink parameters

Each data sink includes a set of parameters to control its operation. The list of parameters vary depending on the data sink you chose to work with.




DigitalEdge validation checks parameters for structure and syntax, but no communication checks are performed with these parameters.

---

You can access data sink parameters two ways:

- To *add* a new data sink and its parameters: **System Builder > Overview** tab > **Data Sinks** section > **Add** > Select a data sink > Double-click a **Parameter's Current Value** to change it, or click the **Data Models** tab to associate another data model with the data sink.
- To *edit* a previously configured data sink, double-click the data sink name on the **System Builder > Overview** tab to access the **Edit Datasink Parameters** dialog box. Double-click a **Parameter's Current Value** to edit it, or click the **Data Models** tab to associate another data model with the data sink.



If you edit parameters for a data sink that is being used in a running system, you must go to the **Management Console** and **Update** the system version that is using that data sink. 

---

Here are detailed lists of parameters, descriptions, and valid values for the data sinks included in the core release. You can also hover over a parameter name in DigitalEdge for tool-tip help.

## Alerting data sink

This data sink does not store DigitalEdge records or alert notifications; it filters processed records for alert triggers and send out alert messages. The alerting data sink specifies how alerts are issued: either as email messages or as messages in a JMS topic. The parameters specify the connection and capabilities of your email server.

AlertingDataSink	
Parameter	Explanation
auth	<p>Whether or not your SMTP email server requires authentication; if set to true, you must supply values for <code>email-from</code> and <code>email-from-password</code></p> <p>Default: false</p>
email-from	<p>If a username is required to connect to the SMTP email server, specify it here.</p> <p>This username is also used as the "From" address in the alert mail messages; must have sending rights to this address</p>
email-from-password	<p>If a password is required to connect to the SMTP email server, specify it here</p>
email-port	<p>The port number that the SMTP email server is running on</p> <p>Default: 25</p>
email-server	<p>Name or IP address of the SMTP email server used to send email alert messages</p> <p>Default: Amazon</p> <p>If you use Amazon's SMTP server, you do not have to specify values for any of the other parameters</p>
send-email	<p>Whether or not alert email messages are sent out; if you select false, you should specify a <code>topic</code> notification instead of email. You can specify both <code>send-email</code> and <code>topic</code> parameters, or just one of the parameters, but you must specify at least one of them.</p> <p>Default: false</p>
tls	<p>Whether or not encryption is required for email messages</p> <p>Default: false</p>
topic	<p>If alerts will be generated as messages on the external JMS queue, you must specify a value for this <code>topic</code> parameter. This value should be a valid JMS topic name, such as <code>com.org.rtw.alert</code>. With a JMS topic, alert messages are posted on one message board for everyone to view.</p> <p>This parameter can have no value, to signify no JMS alerting, or a non-blank value to turn on JMS alerting.</p> <p>You can specify both <code>send-email</code> and <code>topic</code> parameters, or just one of the parameters, but you must specify at least one of them.</p>

AlertingDataSink	
Parameter	Explanation
	Default: null

### Cassandra data sink

Use this beta data sink to store data in an Apache Cassandra cluster. Cassandra is a high-performance open source distributed DBMS that is used to handle big data across many servers with no single point of failure. It is a column-oriented database like HBase. The Cassandra cluster is decentralized, with data distributed across the cluster, but without a master node so that each node can fulfill any request. It is very easy to add nodes to a Cassandra configuration. The Cassandra data sink is not auto-scaling.

CassandraDataSink	
Parameter	Explanation
compression	<p>Indicates how data should be compressed when stored in Cassandra</p> <ul style="list-style-type: none"> <li>• None</li> <li>• LZ4Compressor</li> <li>• SnappyCompressor</li> <li>• DeflateCompressor</li> </ul> <p>Default: None</p>
keyspace	<p>The container for your application data and the Cassandra replication and cluster configuration</p> <p>Recommendation: Do <i>not</i> change the keyspace name once you have created it</p>
replication-factor	<p>Controls the number of data replicas to create</p> <p>Default: 1</p> <p>Recommendation: The default is set to 1 (which results in 2 copies of your data) to conserve storage resources. It is best to set the replication-factor in the range of 1-3.</p>
table-name	The table to use for storing your data

### Dimension data sink

Use this data sink to store data for dimension table enrichments in the tenant database. It provides an alternative method of uploading enrichment data into a dimension table (instead of using the **Table Manager>Batch Tools>Import** functionality to upload CSV data into a table). Your table

columns must first be defined in Table Manager and mapped to your input model before using this data sink to assure that the source data is processed correctly.

You can also use this data sink to store enriched records in the tenant database and use them later as a data enrichment source for a second data feed. In other words, when you have two data sources, one of which will enhance the other data source, configure a dimension data sink. This data sink requires two special fields (tableName and tableKey) in the input model to associate the dimension table and keys with the data model.

DimensionDataSink	
Parameter	Explanation
updates-per-second	<p>The rate at which data is sent into the dimension data sink. This parameter prevents the parser queue from filling up before the data can be processed into the data sink. To help determine an optimum value for this parameter, use <b>System Monitor &gt; Queue Size</b> graph to check on the processing backlog and how many messages were fed in that are not yet processed. The parameter value represents the number of messages fed into the data sink per second.</p> <p>Default: 500</p>

### Elasticsearch data sink

This beta data sink implements the open source Elasticsearch full text search engine. This data sink stores and indexes your data and provides real-time full text querying with a user friendly search interface. Elasticsearch is auto-scaling, auto-connects to other DigitalEdge nodes for true distributed processing, and support multi-tenancy. Each node hosts multiple index shards, forming and managing clustered operations. The Elasticsearch data sink needs no parameters for configuration; it is one of the easiest and quickest data sinks to implement in DigitalEdge, yet provides comprehensive processing and auto-scaling. With Elasticsearch, you can have massive amounts of streaming data ingested, indexed, and searchable in seconds.

### External HBase data sink

Use this data sink to store data in an existing Hadoop/HBase cluster that is external to and not managed by DigitalEdge.

ExternalHBaseDataSink	
Parameter	Explanation
record-write-retry-count	<p>Reserved for future use</p> <p>Default: 20</p>
record-write-retry-delay	Reserved for future use

ExternalHBaseDataSink	
Parameter	Explanation
	Default: 10000
row-key-resolver-name	<p>The method used to determine the rowKey for written records. Change the default only if you need to compute a more meaningful key instead of the random UUID.</p> <p>Default: Enriched_Record_Row_Key_Resolver (UUID)</p>
table-name-resolver-name	<p>The method used to determine the HBase table name for storing enriched records. The default value, <code>Data_Model_Table_Name_Resolver</code>, uses a name determined by your data model. If you select <code>User_Configured_Table_Name_Resolver</code>, enter a table name for enriched records in the <code>user-specified-table-name</code> parameter.</p> <p>Default: <code>Data_Model_Table_Name_Resolver</code></p>
user-specified-table-name	<p>The name of the table used for writing enriched records; required only if the <code>table-name-resolver-name = User_Configured_Table_Name_Resolver</code></p>
write-level	<p>Specifies what parts of JSON records (i.e., the granularity) are writable to the HBase data store. HBase uses one table per data model for storage, and one column family for each parameterized value. For example, with the default of <code>Metadata, Objects, Fields</code>, JSON records are written to HBase three times: <code>metadata</code> to one column family, <code>objects</code> to a second column family, and <code>fields</code> to a third column family; one DigitalEdge record per HBase row. Decisions for write-level values are dependent upon the web and user apps that you specify. For example, committing <code>Fields</code> to HBase may require more processing and storage space, but provides applications with queryable fields and full context data. <code>Metadata</code> storage may help facilitate record matching for a Pentaho dashboard application.</p> <p>Default: <code>Metadata, Objects, Fields</code></p> <p>Recommendation: <code>Metadata, Objects</code></p>
zookeeper-quorum	<p>A comma delimited list of hostnames or IPs of the external ZooKeeper cluster used to communicate with the external HBase cluster. For example: <code>zookeeper1,zookeeper2</code></p>

### External HDFS data sink

Use this data sink to store data in an existing Hadoop cluster that is external to DigitalEdge, is not managed by DigitalEdge, and is compatible with Cloudera CDH3uX releases. You must specify the

communication connection to your organization's Hadoop cluster with these parameters. If a firewall is between DigitalEdge and the external Hadoop data sink, you must open ports for DigitalEdge access.

ExternalHDFSDataSink	
Parameter	Explanation
block-size	Specifies the <code>dfs.block.size</code> for files written to HDFS. The default value is a medium size that can be used as a starting point.  Default: 134217728
namenode-hostname	The IP address or valid, resolvable host name of the server that is running the NameNode daemon in your Hadoop cluster  Default: namenode.domain  Recommendation: keep the default value to use the internal Hadoop cluster
namenode-port	The port number that the NameNode daemon is running on in your Hadoop cluster  Default: 8020  Recommendation: keep the default value to use the internal Hadoop cluster
replication-factor	Controls the default replication factor on files written to HDFS by this data sink
target-folder	The directory where the data sink will write records prior to insertion into Hadoop; must be writable by the user running this data sink in DigitalEdge; can change the default root and/or folder name for your site configuration  Default: /tmp/external_hdfs_data_sink

### External Hive data sink

Use this data sink to store data in an existing Hadoop/Hive environment that is external to DigitalEdge and is not managed by DigitalEdge. You must specify the communication connection to your organization's Hive cluster with these parameters. If a firewall is between DigitalEdge and the external Hive data sink, you must open ports for DigitalEdge access.

ExternalHiveDataSink	
Parameter	Explanation
block-size	Specifies the <code>dfs.block.size</code> for files written to HDFS. The default value is a medium size that can be used as a starting point. Default: 134217728
compress-data	Indicates if data should be compressed before it is stored; only works if Snappy compression is configured on the external Hive cluster Default: false
hdfs-working-folder	The directory where the data sink will write records prior to insertion into Hive; must be writable by the user running this data sink in DigitalEdge; can change the default root and/or folder name for your site configuration Default: /tmp/hiveDataSink
hive-jdbc-hostname	The IP address or valid, resolvable host name for the server running the Hive Thrift service Default: localhost Recommendation: keep the default value to use the embedded server
hive-jdbc-port	The port number for the server that is running the Hive Thrift service Default: 10000 Recommendation: keep the default value to use the embedded server
jobtracker-hostname	The IP address or valid, resolvable host name for the server running the Hadoop JobTracker daemon Default: jobtracker.domain Recommendation: keep the default value to use the internal Hadoop cluster
jobtracker-port	The port number that the Hadoop JobTracker is running on in your Hadoop cluster Default: 8010 Recommendation: keep the default value to use the internal Hadoop cluster
metastore-url	The full JDBC URL to the Hive metastore RDBMS. Use an



ExternalHiveDataSink	
Parameter	Explanation
	<p>IP address unless the hostname is publicly accessible. Suggested format:</p> <p>jdbc:HIVE_METASTORE_JDBC_PARAMETERS//HIVE_METASTORE_IP:HIVE_METASTORE_PORT</p>
namenode-hostname	<p>The IP address or valid, resolvable host name of the server that is running the NameNode daemon in your Hadoop cluster</p> <p>Default: namenode.domain</p> <p>Recommendation: keep the default value to use the internal Hadoop cluster</p>
namenode-port	<p>The port number that the NameNode daemon is running on in your Hadoop cluster</p> <p>Default: 8020</p> <p>Recommendation: keep the default value to use the internal Hadoop cluster</p>
password	The password that Hive uses to connect to the metastore
replication-factor	<p>Controls the default replication factor on files written to the Hadoop Distributed File System by this data sink</p> <p>Default: 2</p>
use-complex-schema	<p>Enables complex table schema definitions (such as map &lt;&gt;, array &lt;&gt;, struct &lt;&gt;, etc.) derived from the canonical data model. This parameter requires the use of a supported SerDe (a Serializer/Deserializer that lets Hive read in data from a table and write it out to HDFS); true or false</p> <p>Default: false</p>
username	The username that Hive uses to connect to the metastore

## HBase

Use this data sink to store DigitalEdge data in an HBase database that is managed internally by DigitalEdge. Configure the parameters to define how records are stored in HBase.

HBaseDataSink	
Parameter	Explanation
record-write-retry-count	Reserved for future use Default: 20
record-write-retry-delay	Reserved for future use Default: 10000
row-key-resolver-name	The method used to determine the rowKey for written records. Change the default only if you need to compute a more meaningful key instead of the random UUID. Default: Enriched_Record_Row_Key_Resolver (UUID)
table-name-resolver-name	The method used to determine the HBase table name for storing enriched records. The default value, <code>Data_Model_Table_Name_Resolver</code> , uses a name determined by your data model. If you select <code>User_Configured_Table_Name_Resolver</code> , enter a table name for enriched records in the <code>user-specified-table-name</code> parameter. Default: <code>Data_Model_Table_Name_Resolver</code>
user-specified-table-name	The name of the table used for writing enriched records; required only if the <code>table-name-resolver-name = User_Configured_Table_Name_Resolver</code>
write-level	Specifies what parts of JSON records (i.e., the granularity) are writable to the HBase data store. HBase uses one table per data model for storage, and one column family for each parameterized value. For example, with the default of <code>Metadata, Objects, Fields</code> , JSON records are written to HBase three times: <code>metadata</code> to one column family, <code>objects</code> to a second column family, and <code>fields</code> to a third column family; one DigitalEdge record per HBase row. Decisions for write-level values are dependent upon the web and user apps that you specify. For example, committing <code>Fields</code> to HBase may require more processing and storage space, but provides applications with queryable fields and full context data. <code>Metadata</code> storage may help facilitate record matching for the Pentaho dashboard application. Default: <code>Metadata, Objects, Fields</code> Recommendation: <code>Metadata, Objects</code>

## Hive

Use this data sink to create a Hive cluster that is managed by DigitalEdge. Recommendation: Keep the default connection values for this data sink, and turn compression on.

HiveDataSink	
Parameter	Explanation
block-size	Specifies the <code>dfs.block.size</code> for files written to HDFS. The default value is a medium size that can be used as a starting point. Default: 134217728
compress-data	Indicates if data should be compressed before it is stored Default: false
hdfs-working-folder	The directory path in HDFS where pipeline-processed data is written and queries are run against it; must be writable by the user running this data sink in DigitalEdge; you can change the default folder name but <i>not</i> the root Default: /tmp/hiveDataSink
hive-jdbc-hostname	The IP address or host name for the server running the Hive Thrift service Default: localhost Recommendation: keep the default value to use the embedded server
hive-jdbc-port	The port number for the server that is running the Hive Thrift service Default: 10000 Recommendation: keep the default value to use the embedded server
jobtracker-hostname	The IP address or host name for the server running the Hadoop JobTracker daemon Default: jobtracker.domain Recommendation: keep the default value to use the internal Hadoop cluster
jobtracker-port	The port number that the Hadoop JobTracker is running on Default: 8010 Recommendation: keep the default value to use the internal Hadoop cluster
namenode-hostname	The IP address or host name of the server that is running the

HiveDataSink	
Parameter	Explanation
	NameNode daemon Default: namenode.domain Recommendation: keep the default value to use the internal Hadoop cluster
namenode-port	The port number that the NameNode daemon is running on Default: 8020 Recommendation: keep the default value to use the internal Hadoop cluster
replication-factor	Controls the default replication factor on files written to the Hadoop Distributed File System by this data sink Default: 2
use-complex-schema	Enables complex table schema definitions (such as map <>, array <>, struct <>, etc.) derived from the canonical data model. This parameter requires the use of a supported SerDe (a Serializer/Deserializer that lets Hive read in data from a table and write it out to HDFS); true or false Default: false

### JSON to JDBC data sink

Use this data sink to grab JSON objects, map them to an SQL database table, and write them to a specified database via JDBC. The data sink requires two special fields (tableName and tableKey) in the input model to associate the dimension table and keys with the data model.

This data sink requires just one parameter, to specify the destination database connection.

JsonToJdbcDataSink	
Parameter	Explanation
connection-url	<p>A pointer to the destination relational database (e.g., the DigitalEdge tenant database), in the format:</p> <pre>jdbc:&lt;databasetype&gt;[:protocol]://&lt;host&gt;[:port]/[database]</pre> <p>For example:</p> <pre>jdbc:mysql://localhost:3306/mytestdb</pre> <pre>jdbc:h2:tcp://localhost:8161/commondb</pre> <p>Default: null</p>

### Lucene indexing data sink

The Lucene data sink builds an inverted index that is optimized for real-time searching of DigitalEdge data. Use these parameters to control the indexing process. This data sink stores index entries, not fully processed records.

The Lucene indexing data sink works with several web apps:

- The Search web app: a search application based on the Solr™ open source enterprise search platform from Apache Lucene™
- The SearchAPI: a REST API for real time searches with the Zoie search engine

If you use either of these search applications, or if you are building a custom search client (for example, a Flex application or a Javascript browser application), you *must* set up a Lucene data sink to index the processed DigitalEdge records.

---

★ When sizing a Lucene data sink (`datasink.lucene`) with the Process Group Parameters, you should allocate 50% extra storage for index building and merging. For example, if you anticipate needing 1 TB space for a Lucene index, configure it for 1.5 TB.

---

LuceneIndexingDataSink	
Parameter	Explanation
always-analyze	<p>List of data fields that should always be tokenized (analyzed, parsed, and prepared for indexing); the field list should be comma-delimited</p> <p>DigitalEdge uses the Lucene StandardAnalyzer for tokenization. Consult the Apache Lucene product documentation for details on how records are analyzed and tokenized.</p> <p>Default: null</p> <p><b>NOTE:</b> When specifying this parameter, you should also set the <code>index-control</code> parameter to either <code>Fields</code> or <code>ContentAndFields</code>.</p>
do-not-analyze	<p>List of data fields that should never be tokenized (analyzed, parsed, and prepared for indexing); ); the field list should be comma-delimited</p> <p>DigitalEdge uses the Lucene StandardAnalyzer for tokenization. Consult the Apache Lucene product documentation for details on how records are analyzed and tokenized.</p> <p>Default: null</p> <p><b>NOTE:</b> When specifying this parameter, you should also set the <code>index-control</code> parameter to either <code>Fields</code> or <code>ContentAndFields</code>.</p>
index-control	<p>Specifies what parts of data records should be indexed:</p> <ul style="list-style-type: none"> <li>• <code>ContentOnly</code>: the entire body of the JSON record</li> <li>• <code>FieldsOnly</code>: fielded data only</li> <li>• <code>ContentAndFields</code>: both the JSON body and record fields</li> <li>• <code>None</code></li> </ul> <p>Double-click to select from the drop-down menu.</p> <p>Default: <code>ContentAndFields</code></p> <p><b>NOTE:</b> If you choose to index <code>FieldsOnly</code> or <code>ContentAndFields</code>, you can optionally specify the <code>always-analyze</code> or <code>do-not-analyze</code> parameter to selectively limit the indexed fields to a specify subset of fields.</p>
zoie-batch-delay	<p>Number of milliseconds between each batch submission to Lucene before in-memory index is flushed</p>

LuceneIndexingDataSink	
Parameter	Explanation
	Default: 60000
zoie-batch-size	<p>Number of fields to store in each batch submission to Lucene before memory is flushed</p> <p>Default: 10000</p> <p>Recommendation: keep the default value</p>

### MongoDB data sink

Use this data sink to build a general purpose MongoDB data store for processed data. You can configure MongoDB once per system. Each instance sets up its own copy of MongoDB; no cluster is created. The parameters specify the connection, timeout, and database name. Note that MongoDB has a maximum record size of 16 MB.

MongoDbDataSink	
Parameter	Explanation
auto-connect-retry	<p>Determines if DigitalEdge times out or keeps looking for data on the connection; must be <code>true</code> for the connect-timeout-ms parameter to work; double-click to select true or false from the drop-down menu</p> <p>Default: true</p> <p>Recommendation: keep the default value</p>
connect-timeout-ms	<p>How long (in milliseconds) DigitalEdge will wait for a connection before it gives up looking for data on the queue</p> <ul style="list-style-type: none"> <li>0 = no timeout; if a connection is not established, an error is returned</li> <li>any number higher than 0 = DigitalEdge will retry looking for data</li> </ul> <p>Default: 0</p> <p>Recommendation: if your network is slow, set this parameter to a higher number; also, set auto-connect-retry parameter to <code>true</code></p>
database-name	<p>Name the DigitalEdge data store with any valid MongoDB name of your choice</p> <p>Default: dbname</p>
mongo-server-host	Name of the host server running MongoDB; multiple server names

MongoDbDataSink	
Parameter	Explanation
	should be separated by spaces or tabs Default: localhost server1 server2 Recommendation: localhost
mongo-server-port	Dedicated port on which MongoDB will run; must be between 1024 and 65535 Default: 27017

### Sleep data sink

This data sink is used strictly for test purposes. It reads a record, optionally creates a log entry, and then sleeps for a specified amount of time, to test the integrity of the DigitalEdge pipeline. It does not store or process records.

SleepDataSink	
Parameter	Explanation
delay	Time between reads while processing records; expressed in milliseconds Default: 5000
input-logging	Optionally creates an entry in the ingest log for each record read Default: false

### Sending data to an external application


DigitalEdge processed data can be used by applications outside of DigitalEdge. For example, you may have a reporting application that is standard in your organization, and you would like to generate DigitalEdge analytical reports in a familiar format.

You can configure a scripting data sink to feed data from DigitalEdge into the reporting application. This data sink uses a JSR-223 compliant script (Groovy, JavaScript, etc.) to transport data to an external application.

Alternately, you could also use one of the data sinks that are external to DigitalEdge: the external Hive, HBase, or HDFS data sink.

1. Open **System Builder**.
2. Go to the **Overview** tab.
3. Use the **Data Sinks** section.




4. Click **Add**.
5. Choose the **ScriptingDataSink** from the **Select a Data Sink** dialog box and click **OK**.
  - a. Edit the default values in the **Set Datasink Parameters** dialog box:
    - **engine-name**: name of the scripting engine, in the form of a valid JRE name as specified by JSR-223
    - **parameter**: a list of parameters required by your script, delimited by the character used by your scripting language
    - **script-file**: the full path of the script to run
  - or
  - b. Choose the **ExternalHBaseDataSink**, **ExternalHdfsDataSink** or the **ExternalHiveDataSink** and click **OK**. [See "Data sink parameters" on page 66](#) for details about configuring their parameters.
7. Click **OK**.
8. Click **Build**. 

## Fine-tuning auto-scaling

You can adjust the number of volumes and their sizes assigned to any processor group in your system. Reallocating resources may help with auto-scaling performance issues.


---

 Each process group is associated with a pre-defined DigitalEdge security group. Security groups are managed in the **Management Console** on the **Security** page. You must have security group permissions to edit parameters here.

---

 Not all auto-scaling parameters are editable. Grayed-out values cannot be adjusted.

---

1. Open **System Builder**.
2. Go to the **Details** tab.
3. Select a process that needs adjustment.
4. Edit parameters that specify your auto-scaling needs
  - a. Double-click on the value in the **Inst Size** or **Scale?** cell and select a new value from the allowable list.
  - b. Double-click on a value in the numeric parameter fields and edit the value:
    - **Min**
    - **Max**
    - **Alloc**
    - **Dealloc**
5. Click **Build**. 

## Resizing a process or server

You can adjust the number of volumes and their sizes assigned to any processor group in your system. Reallocating resources may help with auto-scaling performance issues.

---

★ Each process group is associated with a pre-defined DigitalEdge security group. Security groups are managed in the **Management Console** on the **Security** page. You must have security group permissions to edit parameters here.

---

1. Open **System Builder**.
2. Go to the **Details** tab.
3. Select a processor group to work with.
4. Double-click on the value in the **# Volumes** cell and select a new value from the allowable list.

---

★ The **# Volumes** parameter cannot be changed after you **Start** a system.

---

5. Double-click on the value in the **Vol Size (GB)** cell and edit the value.
6. Click **Build**. ✕

See the DigitalEdge *Configuration Guide* for process group parameter details.)

## Process group parameters

A process group represents a category of processors, not individual instances of processors, representing one functional area (transport, ingest, index, etc.). You can view process group parameters in **System Builder**, on the **Details** tab. Process groups are built based on the choices you made for components and applications on the **Overview** tab of the **System Builder**. For example:

- ingest.all
- transport
- datasink.mongodb
- jms.internal

Process group parameters control auto-scaling and resource allocation of each instance (VM) in the group. Default parameter values are initialized based on the **System Size** that you chose on the **Overview** tab.

---

★ Each process group is associated with a pre-defined DigitalEdge security group. Security groups are managed in the **Management Console** on the **Security** page. You must have security group permissions to edit parameters here.

---

Each process group shares a set of parameters which are applied to each instance:

Parameter	Explanation
# Volumes	How many volumes are currently consumed by an instance of this process group (cannot be changed after you <b>Start</b> a system)
Vol Size (GB)	Size per volume, in gigabytes
Public IP	<p>Whether or not the process uses a persistent IP. You can assign an IP address by double-clicking on this value. In the <b>Specify Persistent IP Address</b> pop-up, either enter a <b>Specific Address</b> or click <b>Allocate IP Address</b> for DigitalEdge to assign an address.</p> <p>The only time you <i>must</i> assign an IP is for the webapps.main node if you are running on a Eucalyptus platform and you do not have an external DNS forwarder configured. If you are running in VPC, you do not have to assign IP addresses; AWS assigns elastic IP addresses automatically.</p>
Inst Size	Size of an instance, as defined by Amazon's node instance types (Xsmall, small, medium, large, Xlarge, XXlarge)
Min	The initial allocation of nodes to an instance is expressed as a range from minimum to maximum; this is the minimum number
Max	<p>The initial allocation of nodes to an instance is expressed as a range from minimum to maximum; this is the maximum number.</p> <p>Set the <b>Max</b> parameter if you need to cap auto-scaling to the terms of a public cloud services contract.</p>
Alloc	When auto-scaling up an instance, nodes are added in quantities; this is the number of nodes that are allocated each time the process is scaled up
Dealloc	When auto-scaling down an instance, nodes are removed in groups rather than individually; this is the number of nodes that are deallocated each time the process is scaled down. Use caution when deallocating instances; in some case, you may lose data when you scale down (especially in Hadoop clusters).
Scale?	Whether or not a process is auto-scaling
JMS Store	Turn data persistence on or off for an instance; primarily applies to the internal instances which keep data in memory and do not persist data to disk by default; the JMS external node is always persisted to disk; valid values = true or false

★ When sizing a Lucene data sink (`datasink.lucene`), you should allocate 50% extra storage for index building and merging. For example, if you anticipate needing 1 TB space for a Lucene index, configure it for 1.5 TB.

### Data Sink Parameters panel

Use this panel to modify the parameter values that were previously specified in the **Set Datasink Parameters** dialog box.

### Transport Parameters panel

Use this panel to modify the parameter values that you specified in the **Set Transport Parameters** dialog box.

## Opening a port for a new component


You can open a new port for an additional DigitalEdge component at any time.

1. Add the new component in **System Builder**.
2. Use the **Management Console** to specify a security rule and the port settings:
  - a. Select **Security** > [select a security group] > **Add Security Rule**. The **Add New Security Rule** dialog box prompts for details.
  - b. Select a **Protocol** from the drop-down menu (TCP, UDP, or ICMP).
  - c. Enter a **Port Range** (one port number or a range of ports).
  - d. Enter a communications **Source**. A source can be an IP address, a range of IP addresses, or another security group that can talk with the assigned process groups.
  - e. Click **ADD**.

## Assigning IP addresses

You rarely have to assign IP addresses to nodes.

### Gateway

You can view the IP addresses assigned to the gateway node in **Management Console** by double-clicking the **Gateway** node  Gateway | CPU Resources Used: 69% .

### AWS™ EC2

An Amazon system is automatically assigned public IP addresses. You can view the IP addresses in: **Management Console** > **Systems** > select a system > **Process Groups** tab > double-click a process **Group Name**.

System: JmsTest8.rtsaic.com   Process Group: ingest.all				
<div> <div>Actions</div> <div>Copy</div> </div>				
Name	Instance ID	Persistent IP	Public IP	Private IP
ingest-all-node1.JmsTest8.rtsaic.com	i-cdeb8aec	-	23.22.151.20	10.77.79.47

Use the **Copy** function whenever you need to copy/paste an IP address, instance ID, or instance name to another application. Highlight a **Name** row, click **Copy**, and select an item you want to copy to the clipboard. You can then paste that attribute from the clipboard.

## AWS™ VPC

Elastic IP addresses are automatically assigned in a VPC system. You can view the IP addresses in: **Management Console > Systems > select a system > Process Groups tab > double-click a process Group Name.**

System: JmsTest8.rtsaic.com   Process Group: ingest.all				
<div> <div>Actions</div> <div>Copy</div> </div>				
Name	Instance ID	Persistent IP	Public IP	Private IP
ingest-all-node1.JmsTest8.rtsaic.com	i-cdeb8aec	-	23.22.151.20	10.77.79.47

Use the **Copy** function whenever you need to copy/paste an IP address, instance ID, or instance name to another application. Highlight a **Name** row, click **Copy**, and select an item you want to copy to the clipboard. You can then paste that attribute from the clipboard.

## Eucalyptus

If you configured an external DNS forwarder, a Eucalyptus-based system does not need IP addresses.

If you did *not* configure an external DNS forwarder to work with a Eucalyptus-based DigitalEdge system, you *must* assign a persistent IP address to the **webapps.main** node. Go to **System Builder > Details > highlight the webapps.main Group Name > double-click a Public IP value.** In

the **Specify Persistent IP Address** pop-up, either enter a **Specific Address** or click **Allocate IP Address** for DigitalEdge to assign an address.

Failure to assign an IP address to webapps.main will spawn a validation error message in **System Builder** when you attempt to **Build** a new or edited system configuration.




## Transport parameters

Each pre-existing transport includes a set of parameters to control its operation. The list of parameters may vary depending on the parser you chose to work with.

You can access transport parameters two ways:

- To *add* a new transport and its parameters: **System Builder** > **Overview** tab > **Transports** section > **Add** > Select a transport > Double-click a **Parameter's Current Value** to change it
- To *edit* transport parameters, double-click the transport name on the **System Builder** > **Overview** tab to access the **Edit Transport Parameters** dialog box. Double-click a **Parameter's Current Value** to edit it.

★ If you edit parameters for a transport that is being used in a running system, you must go to the **Management Console** and **Update** the system version that is using the transport. 

Here are detailed lists of transport parameters, descriptions, and values for the transports included in the core release. You can also hover over a parameter name in DigitalEdge for tool-tip help.

## DatabaseWatcherTransportService

The Database Watcher transport is a specialized polling service that gets all the data from a database and pulls it into DigitalEdge by periodically running an SQL select query against a database. The database can be queried regularly, starting at the point where the query last left off. So, when the query is run again, only records not selected in the previous query will be retrieved. An

S3 bucket is used to store a backup copy of the data file. To use this transport, you must specify several parameters that define the SQL query, the column that serves as the key/identity column, the stopping point, and how often the query should be run.

DatabaseWatcherTransportService	
Parameter	Explanation
bucket-name	The name of the Amazon S3™ bucket to store a backup copy of the incoming data file. The name must match exactly the name as it is listed in the AWS™ Management Console bucket list
file-key	This parameter is the name of a file that will be placed in the S3 bucket (specified in the bucket-name parameter). This file stores the highest memory-key-column value retrieved in the latest query run (as a backup). By storing the highest value previously read, the transport assures that when the query is run again, only records not selected in the previous query will be retrieved.
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
memory-key-column	The memory-key-column parameter specifies which column number to read in the SQL query to identify new records that were ingested since the last read. The column is 1-indexed. The memory-key-column acts as a surrogate key to indicate how far the query gets into the database on each run.  Default = 1
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0
select-data-statement	The SQL query that is run against the database
sleep-time	The amount of time, in milliseconds, between polling the S3 bucket for data.  Default = 0

## DirectoryCrawlerTransportService

The Directory Crawler beta transport is similar to the Directory Watcher transport, processing data in a local or remote file system. But it includes additional capabilities beyond that of the Directory Watcher. The Directory Crawler transport decompresses any zipped files and processes files that match wild card patterns. This transport is also recursive on the source-directory; it will crawl subfolders.

Compression formats currently supported by the transport include:

- Zip (\*.zip)
- Tar (\*.tar)
- Tar-Gzip (\*.tar.gz, \*.tgz)
- Gzip (\*.gz)
- 7-Zip (\*.7z)
- RAR (\*.rar)
- Tar-Bzip2 (\*.tbz2, \*.tar.bz2)
- Bzip2 (\*.bz2)

DirectoryCrawlerTransportService	
Parameter	Explanation
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
map-input-models	<p>Specify file mappings in the following format:</p> <pre>fileName1:recordFormat1:recordHeaderLines1:inputFormat1, fileName2:recordFormat1:recordHeaderLines2:inputFormat2, fileName3:recordFormat3:recordHeaderLines3:inputFormat1</pre> <p>where:</p> <p>fileName can include wildcards:</p> <ul style="list-style-type: none"> <li>• ? represents a single character</li> <li>• * represents multiple characters</li> </ul> <p>recordFormat is the data record type to determine how to split multiple records on the appropriate boundaries</p> <p>recordHeaderLines is the number of header lines that can be stripped out of the records</p> <p>inputFormat must match a Data Source name as defined in the Data Model Editor</p> <p>White spaces are ignored.</p>



DirectoryCrawlerTransportService	
Parameter	Explanation
record-format	<p>A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu</p> <p>Default = NULL</p> <p>Recommendation: Use the BASE64_CONTENT record format when also using the UnstructuredFileParser in the same system</p>
record-header-lines	<p>How many header lines should be stripped out of the data records</p> <p>Default = 0</p>
remote-server	<p>If you are reading data from a remote server (not a local system), this parameter specifies the IP address or domain name of the remote NFS server.</p>
remote-share-name	<p>If you are reading data from a remote server (not a local system), this parameter specifies the name of the shared directory on the remote-server from which files are read and fed into the transport. DigitalEdge uses the source-directory parameter as the local mount point to this shared directory.</p>
source-directory	<p>The local directory path to crawl for data, expressed in Linux style notation, not Windows notation. If you have also defined the remove-server and remote-share-name parameters, DigitalEdge will create this directory and use it as the local mount point.</p> <p><b>NOTE:</b> This value must be unique if you have multiple transports using a remote-server option in your DigitalEdge system.</p>

### DirectoryWatcherTransportService

The Directory Watcher transport is a polling service similar to a polling S3 File Transport, except that the transport is watching a file system, not an Amazon S3™ bucket. This transport is typically used when your data files are local and you do not want to move them to S3™. But the transport can also watch a directory on a remote server. To use this transport, you must specify the directory path and the polling time interval. The transport watches one directory folder; it is not recursive.



The Directory Watcher Transport does not unzip and process zipped files. You should extract and unzip these files if you want this transport to ingest that data content.

DirectoryWatcherTransportService	
Parameter	Explanation
check-interval	How often the transport looks at the directory for new data, in milliseconds  Default = 500
content-encoding	Indicates if the transport will encode received bytes before submitting them for processing <ul style="list-style-type: none"> <li>• None = no encoding will be performed</li> <li>• Base16 = hexadecimal encoding</li> <li>• Base64</li> </ul> Default = None
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
preserve-file-name	Indicates (true or false) if the filename is included in the message sent to the ingest pipeline. Use this parameter in conjunction with the Unstructured File Parser, when you include the data source's filename in the data model's output.  Default = false  Note: This parameter only works when content-encoding is set to Base16 or Base64.
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0
remote-server	If you are reading data from a remote server (not a local system), this parameter specifies the IP address or domain name of the remote NFS server.
remote-share-name	If you are reading data from a remote server (not a local system), this parameter specifies the name of the shared directory on the remote-server from which files are read and fed into the transport. DigitalEdge uses the watched-directory parameter as the local mount point to this shared directory.

DirectoryWatcherTransportService	
Parameter	Explanation
watched-directory	<p>The local directory path to watch, expressed in Linux style notation, not Windows notation. If you have also defined the remove-server and remote-share-name parameters, DigitalEdge will create this directory and use it as the local mount point.</p> <p><b>NOTE:</b> This value must be unique if you have multiple transports using a remote-server option in your DigitalEdge system.</p>

### HiveTransportService

The Hive transport is a specialized transport that gets data from an existing Hive data sink and pulls it into another data sink, either in the same DigitalEdge system or another DigitalEdge system. For example, you might store enriched data in Hive and then transport it to a Lucene data sink in the same system for indexing. Or, you might store enriched data in Hive in one system and then transport it to another DigitalEdge system for iterative enrichment. Also, you would usually create an SQL select query to run against the Hive data sink to pull out a subset of data rather than copying all the data.

HiveTransportService	
Parameter	Explanation
hive-host	<p>The server where the Hive data sink resides, identified as an IP address or DNS name</p> <p>Default = localhost</p>
hive-port	<p>The port number to connect to the hive-host</p> <p>Default = 10000</p>
input-format	<p>Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models</p>
poll-interval	<p>How often the transport looks at the data sink for new data, in milliseconds</p> <p>Default = 60000</p>
record-format	<p>A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu</p> <p>Default = NULL</p>
record-header-lines	<p>How many header lines should be stripped out of the data records</p>

HiveTransportService	
Parameter	Explanation
	Default = 0
select-data-statement	The SQL select query to run against the Hive database. If blank, the transport will take all data, which may cause performance issues.
state-store-mode	Indicates if the position state should be stored locally on the instance; used when the transport goes down, to restart where it left off without duplicating any records  Default = true

### JMSBridgeTransportService

The JMS Bridge Transport copies data directly from an external JMS messaging system to the DigitalEdge JMS server. This is a one-to-one transport, from one JMS queue to another.

JMSBridgeTransportService	
Parameter	Explanation
incoming-address	The address (hostname and port) of the corporate JMS queue (in JMS URL format) from which messages will be fetched and transported. For example:  ssl://jms-node1.dm-test-myorg.com:61616  where: <ul style="list-style-type: none"> <li>• ssl = the transport type</li> <li>• the address name is a single connection to one node, or a failover address</li> <li>• port number = the provider of the messaging service for the queue</li> </ul> Default = localhost:61616
incoming-user	The username needed to connect to the corporate server (if necessary)
incoming-password	The password credentials required to connect to the corporate server (if necessary)
incoming-queue	The name of the corporate queue to connect to. For example: com.myorg.data.parse
incoming-topic	The name of the corporate topic to pull messages for transport
input-format	Identifies the data source and parser format that the transport uses

JMSBridgeTransportService	
Parameter	Explanation
	to pull data off the input queue; double-click to select a Data Source from your specified data models
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0
sleep-time	The amount of time, in milliseconds, between polling of the data.  Default = 1000

### MongoDBTransportService

The MongoDB transport is a specialized transport that gets data from an existing MongoDB data sink and pulls it into another data sink, either in the same DigitalEdge system or another DigitalEdge system. For example, you might store enriched data in MongoDB and then transport it to a Lucene data sink for indexing. Or, you might store enriched data in a MongoDB data sink in one system and then transport it to another data sink in a second DigitalEdge system for iterative enrichment. Also, you would usually create an SQL select query to run against the MongoDB data sink to pull out a subset of data rather than copying all the data.

MongoDBTransportService	
Parameter	Explanation
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
mongo-collection	The name of the MongoDB table to pull data from
mongo-db	The name of the MongoDB database to pull data from  Default = mydb2
mongo-host	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a data model from the drop-down menu
mongo-port	The port number to connect to the mongo-host  Default = 27017

MongoDBTransportService	
Parameter	Explanation
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0
select-data-statement	The SQL select statement to run against the MongoDB database. If blank, the transport will take all data, which may cause performance issues.
state-store-mode	Indicates if the position state should be stored locally on the instance; used when the transport goes down, to restart where it left off without duplicating any records  Default = true

### PcapSnifferTransportService

The pcap sniffer transport captures and splits pcap packets on a network interface that you specify. You can also optionally filter out data from the transport. The pcap transport is often used with the DNS PCAP and SNMP PCAP parsers.

PCapSnifferTransportService	
Parameter	Explanation
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models.
interface-name	Identifies the type of network interface that the transport sniffs for pcap packets; for example, use eth0 for an Ethernet connection.  Default = eth0
pcap-filter	Filters the pcap captures to include or exclude different types of data; for example, you can capture data coming from a specific IP address. See <a href="http://linux.die.net/man/7/pcap-filter">http://linux.die.net/man/7/pcap-filter</a> for a list of filters and the syntax.
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple

PCapSnifferTransportService	
Parameter	Explanation
	records; double-click to select PCAP from the drop-down menu. The PCAP splitter type outputs Base 64 encoded data only.  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0

### S3FileTransportService

The S3 File Transport checks an Amazon S3™ bucket for data ready to transmit to the JMS external queue, and optionally deletes files after processing them. You can configure this transport to poll an S3™ bucket regularly, or to read S3™ just once as soon as the file appears in the bucket. When configured to poll the S3™ bucket periodically, the transport may locate multiple files over time with the same name. To use this transport, you must specify the S3™ bucket, file parser type that DigitalEdge will be using for data input, and several other parameters.

S3FileTransportService	
Parameter	Explanation
bucket-name	The name of the Amazon S3™ bucket to check for input data files. The name must match exactly the name as it is listed in the AWS™ Management Console bucket list.  For example: sales.test.data.
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
polling-interval	How often the transport looks at the S3™ bucket for new data, in milliseconds. Use 0 to indicate that the bucket should be polled just once for data.  Default = 0
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0

S3FileTransportService	
Parameter	Explanation
should_delete_source	Once a file is read and processed, you can optionally delete the file; double-click to select Yes or No.  Default = No

### TCPTransportService

The TCP transport reads data from an entire TCP stream. To use this transport, you must specify the listening port and the file parser format that DigitalEdge will be using.

TCPTransportService	
Parameter	Explanation
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
listen-port	The number of the port to connect to for receiving TCP messages  Default = 0
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0

### TwitterFilterTransportService

This transport gets Tweets from the Twitter feed based on criteria that you define using the Twitter Search API (see <https://dev.twitter.com/docs> for Twitter API documentation). You can search for keywords and/or Twitter usernames. This is the most flexible and commonly used transport of the three Twitter transports. You must have a Twitter account to use this transport.



TwitterFilterTransportService	
Parameter	Explanation
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
o-auth-access-token	Your OAuth access token; see the <a href="#">Twitter API documentation</a> for information about obtaining a token
o-auth-access-token-secret	Your OAuth access token secret; see the <a href="#">Twitter API documentation</a> for information about token credentials
o-auth-consumer-key	Your OAuth consumer key; see the <a href="#">Twitter API documentation</a> for information about obtaining credentials
o-auth-consumer-secret	Your OAuth consumer secret; see the <a href="#">Twitter API documentation</a> for information about obtaining credentials
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0
terms	A comma-separated list of OR'ed keywords you want to search for in Tweets; maximum allowed = 400 words  Hits are AND'ed with the usernames
usernames	A comma-separated list of OR'ed Twitter names you want to follow; maximum allowed = 5000  Hits are AND'ed with the terms

### TwitterRESTTransportService

You must have a Twitter account to use this transport, which follows the Tweets of one Twitter user. The transport uses the Twitter REST API (see <https://dev.twitter.com/docs/api> for Twitter REST API documentation).

TwitterRESTTransportService	
Parameter	Explanation
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu  Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0
username	The name of a Twitter user you want to follow

### TwitterSampleTransportService

This simple Twitter transport selects a random sample of Tweets that you are allowed to read in the Twitter feed. You must have a Twitter account to use this transport. The transport uses the Twitter Search API and the OAuth authentication protocol. See <https://dev.twitter.com/docs> for Twitter API documentation.

TwitterSampleTransportService	
Parameter	Explanation
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
o-auth-access-token	Your OAuth access token; see the <a href="https://dev.twitter.com/docs">Twitter API documentation</a> for information about obtaining a token
o-auth-access-token-secret	Your OAuth access token secret; see the <a href="https://dev.twitter.com/docs">Twitter API documentation</a> for information about token credentials
o-auth-consumer-key	Your OAuth consumer key; see the <a href="https://dev.twitter.com/docs">Twitter API documentation</a> for information about obtaining credentials
o-auth-consumer-secret	Your OAuth consumer secret; see the <a href="https://dev.twitter.com/docs">Twitter API documentation</a> for information about obtaining credentials
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu

TwitterSampleTransportService	
Parameter	Explanation
	Default = NULL
record-header-lines	How many header lines should be stripped out of the data records  Default = 0

### UDPTransportService

The UDP transport captures datagram packets in a UDP stream sent to a configured port. To use this transport, you must specify the maximum packet size, the listening port, and the file parser format that DigitalEdge will be using.

UDPTransportService	
Parameter	Explanation
content-encoding	Indicates if the transport will encode received bytes before submitting them for processing <ul style="list-style-type: none"> <li>• None = no encoding will be performed</li> <li>• Base16 = hexadecimal encoding</li> <li>• Base64</li> </ul> Default = None
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
listen-port	The number of the port to connect to for receiving UDP packets  Default = 0
max-packet-size	Specifies the maximum packet size, to ensure that the buffer is large enough to hold complete messages  Default = 65535
message-processor-class	Name of the Java class that will process the packets  Default = com.saic.rtw.transport.Services.utils.SimpleUDPMessagesProcess  Recommendation: Keep the default value
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple

UDPTransportService	
Parameter	Explanation
	records; double-click to select a format from the drop-down menu Default = NULL
record-header-lines	How many header lines should be stripped out of the data records Default = 0

### URLTransportService

The URL transport reads the contents of a URL just once and puts it on the JMS input queue. This transport is typically used for pulling data from an RSS feed or from any service that pulls resources from another organization or data source into your system. To use this transport, you must specify the URL and the file parser format that DigitalEdge will be using.

URLTransportService	
Parameter	Explanation
input-format	Identifies the data source and parser format that the transport uses to pull data off the input queue; double-click to select a Data Source from your specified data models
read-address	The URL to read incoming data from
record-format	A data record type (BASE64_CONTENT, PCAP, JSON, TEXTLINE, NULL, TEXTLINEWITHQUOTES) that helps to determine record boundaries when input data includes multiple records; double-click to select a format from the drop-down menu Default = NULL
record-header-lines	How many header lines should be stripped out of the data records Default = 0
user-id	The username needed to validate for https (if necessary)
user-password	The password needed to validate for https (if necessary)

## Chapter 8: Creating Alerts

Many DigitalEdge systems need an active alerting engine to enable use cases such as identifying possible cybersecurity breaches, situational anomalies, and potential threats. Implementing alerts involves:

- Building a system in **System Builder** with three components:
  - **alertcontroller** webapp: to create and manage alert criteria, notifications, and subscriptions
  - **alertsapi** webapp: the back-end component that manages alert criteria
  - **AlertingDataSink**: to filter processed records for alert triggers and to send out alert notifications as email messages or messages in a JMS topic
- Identifying alerting criteria, the business rules that define a potential threat, and translating those rules into data specifications that will trigger an alert
- Writing the messages that should be sent out as alert notifications
- Identifying the people who should be notified of an alert situation

DigitalEdge supports a dynamic alerting infrastructure. The alert model is subscription oriented, so that each user can subscribe to specific alerts. DigitalEdge can also disseminate alerts through mechanisms such as email messages, text messages, mobile devices, or a JMS queue.

### Alerting criteria

The first step in setting up alerts is to define the business rules that identify problematic situations. For example, a bank may want to know if one account is always serviced by one teller, which may indicate potential fraud. Or a network administrator may want to know immediately when an intrusion detection system (IDS) has sensed a possible incident, which may indicate a cybersecurity breach. Business rules are based on a *pattern* that is detected in the *data model*. You define the conditions under which a transaction or a pattern of activity is considered a threat. When data is ingested and processed, the DigitalEdge alert engine filters records against the data rules. Alerting filters are built with the **Alert Controller** and the **Alerts API**. [See "Specifying alerting criteria" on page 102.](#)

### Alert notifications

After you have specified the rules that identify potential fraud situations, you should create a notification that will be sent to key decision makers when an event occurs. When a match is found between a processed record and the alert filter rules, a notification is generated and sent to one or more individuals. Notifications are sent within seconds of DigitalEdge identifying a potential fraud situation. Notifications can be sent via email, a mobile device, or a JMS queue message. Users "subscribe" to alert email notifications; they sign up to receive alert messages. Use the **Alert Controller** to manage notifications and subscriptions. [See "Managing alert notifications" on page 105.](#)

## Alerting data sink

The alerting data sink specifies how alerts are issued. This data sink does not store DigitalEdge records or alert notifications; it filters processed records against the alerting criteria and sends out alert messages. [See "Building an alerting system" on page 102.](#)

## Building an alerting system

To build an alerting system, you must configure three components:

- **alertcontroller** webapp: The **Alert Controller** assists with several alerting tasks:
  - Define the data criteria that trigger anomalous situations
  - Create alert notification messages
  - Define alert subscriptions
- **alertsapi** webapp: The Alerting API webapp is a REST API used to manage the alerting criteria that serve as data triggers for alert notifications. You can use the **Alert Controller** as a front-end UI to the Alerts API.
- **AlertingDataSink**: This data sink filters processed records against alert triggers and sends out alert messages to subscribing decision makers. This data sink does not store DigitalEdge processed records or alert notifications.

Follow these steps in **System Builder**:

1. In the **System Builder**, click the **Overview** tab. Use the **Data Sinks** section.
2. Click **Add**. The **Select a Data Sink** dialog box appears.
3. Click the **AlertingDataSink**.
4. Click **OK**. The **Set Datasink Parameters** dialog box appears. Review the default values and edit any parameter as needed. ([See "Data sink parameters" on page 66](#))
5. Click **OK** to save the parameter values.
6. Next, use the **Webapps/REST APIs** section.
7. Click **Add**. The **Select a Webapp/REST API** dialog box appears.
8. Click the **alertcontroller** webapp.
9. Click **OK**.
10. Click **Save** when you are done with System Builder.



When you Add the **alertcontroller** webapp, DigitalEdge also automatically adds the **alertsapi** webapp to your system.

---

## Specifying alerting criteria

To set up an alerting service in your DigitalEdge system, you must first define the data criteria that trigger anomalous situations. For example, a credit union may want to know when an account

withdrawal is over \$1,000 in a single transaction. Alert rules are based on a *pattern* that is detected in your *data model*. You define the conditions under which a transaction is considered a problem. When data is processed, the Alerting Data Sink engine filters records against these data rules to immediately identify problematic activity.

The **Alerts API** webapp is a REST API used to specify the alerting criteria that serve as data triggers for alert notifications. You can create JSON expressions to run on one or more data model fields. For example, a network security officer may want to flag a user login that originates outside of the local intranet. You define alerting criteria that are specific to your data model and your business needs. See the DigitalEdge *Alerts API Guide* for more details.

Use the **Alert Controller** webapp as a front-end UI interface to the Alerts API. The **Alert Controller** application helps you create *alert definitions*, each of which includes:

- **Alert Criteria:** to define the alerting business rules
- **Email Template:** to notify key staff about an alert situation

The **Alert Criteria** tab helps you define the business rules that trigger an alert situation.

### Using the Alert Controller to create alert criteria

1. Access the **Alert Controller** URL site that your DigitalEdge Administrator has established for the alerting application, such as:


```
https://default.<system_domain_name>/alertcontroller
```

where `system_domain_name` is the full **Domain Name** created in **System Builder**.

Provide your **Username** and **Password** to **Login**. The **Alert Controller** screen appears.

The left panel lists the **Name** of each alert definition that you create, and the **Data Model** that is used to specify that alert filter.


The right panel is the work space for creating **Alert Criteria** and **Email Templates**.

2. Click **New**  to create an alert definition. The **New Alert Definition** screen appears.
3. In the **Name** text box, enter an alert name for the new alerting criteria.

---

★ The maximum length for a **Name** is 125 characters.

---


4. From the drop-down **Data Model** menu, select a data model to work with. This is the enriched data model; all fields are available for building alert criteria.
5. On the **Alert Criteria** tab, enter a JSON expression to define the new alert. (Click **Help**  for reference tips, or see the [JSON](#) web site for instructions on creating a JSON expression.) In general, the JSON expression should include:
  - a. A data model field to work with

- b. A data operator (=, !=, >, <, >=, <=, LIKE, IN, BETWEEN)
- c. A data value to match or filter on

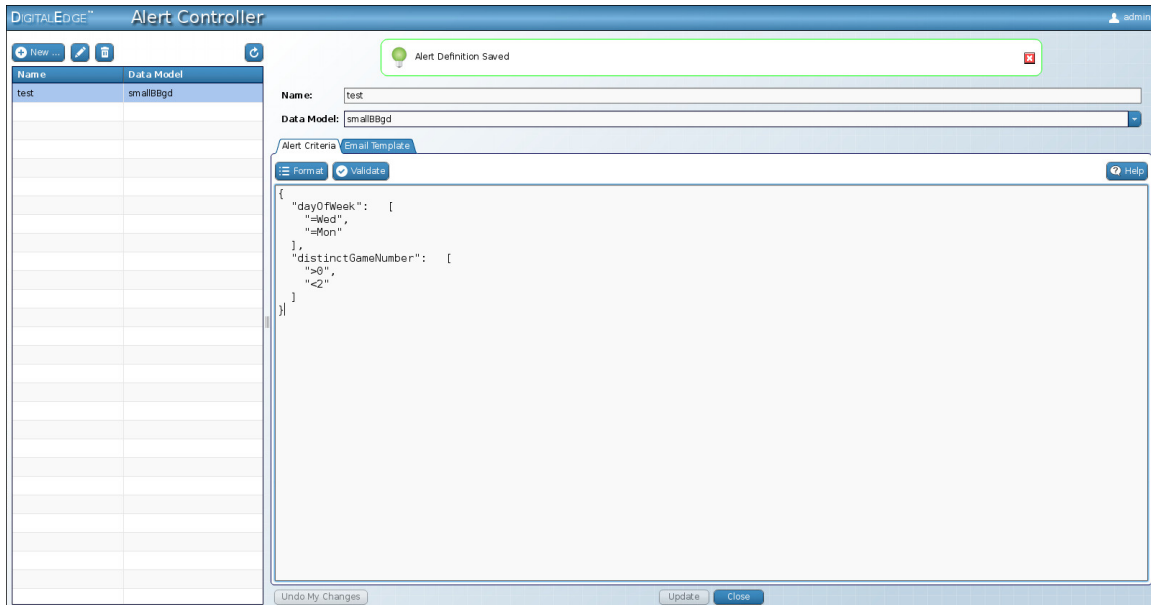
**EXAMPLES:**




- Simple equation: {"AccountID": "=01938"}
- Greater than operator: {"Time": ">'07/24/97 09:14:32'"}
- Less than or equal to operator: {"NumGeos": "<=4"}
- Match values with wildcard characters: {"eventType": "LIKE '\*ellipse'"}
- Compare a field to several different values: {"NumGeos": "IN (4,10,15,2)"}
- Compare the values in two fields: {"Teller":{"Name": "@Account.Name"}}
- Match values that fall between two values, in a range: {"NumGeos": "BETWEEN [4,10]"}
- Multiple criteria: {"eventType": "'Checkpoint'", "extrinsic\_Checkpoint": {"plateNum": "'ABC-1234'"}}
- Nested criteria: [{"networkData": {"source": {"dshield": "LIKE '\*'"}}}, {"networkData": {"destination": {"dshield": "LIKE '\*'"}}}]

**TIPS:**

- Multiple criteria definitions under one **Name** are AND'ed together.
- To OR multiple criteria together, create multiple **Names**, one definition per alert.
- Use @ as a variable to reference the value in a data model field ('@field\_name'). This operator is useful for comparing field values of the same data type.
- The IN operator must have values enclosed in parentheses
- Use wildcards in a **LIKE** expression to match any character:
  - ? = single character wildcard
  - \* = multiple characters wildcard
- String and date string values must be enclosed in single quote marks
- With the **BETWEEN** operation, use parentheses ( ) to indicate exclusivity, brackets [ ] to indicate inclusivity.
- Click **Help**  for a more detailed explanation of the JSON criteria syntax examples.





6. Optionally click **Format**  to display the expression in a more readable JSON pretty-print format.
7. Click **Validate**  to check the syntax of your JSON expression.
8. Click **Create**  to save the criteria to the alert **Name** list.

Or, click the **Email Template** tab to create an email notification for this alert.



You can also highlight an alert **Name** to **Edit**  or **Delete**  it.

## Managing alert notifications

The **Alert Controller** application helps you create alert definitions which include:

- **Alert Criteria:** to define the alerting business data rules
- **Email Template:** to notify staff about an alert situation

The **Email Template** tab helps you:

- Manage email notifications for specific alert criteria
- Create a subject line for an alert notification
- Compose the body of the email message for an alert notification
- Assign email notifications to subscribers (i.e., usernames in DigitalEdge)

## Using the AlertController webapp to manage alert notifications


1. Access the **Alert Controller** URL site that your DigitalEdge Administrator has established for the alerting application, such as:

`https://default.<system_domain_name>/alertcontroller`

where `system_domain_name` is the full **Domain Name** created in **System Builder**.



Provide your **Username** and **Password** to **Login**. The **Alert Controller** screen appears.

2. To create an email notification for a new alert definition that you are specifying, click the **Email Template** tab.

Or, to work with an existing alert definition that you have saved to the alert **Name** list, highlight an alert definition **Name** in the left panel and click **Edit** .

3. Use the **Email Template** tab to define one email notification per alert. A notification consists of a subscriber list, the subject line, and a message.
4. By default, email messages are defined to notify the person who is logged in to the **Alert Controller** as the subscriber. To have alert messages sent to your email address, check the box next to **Me** in the **To** line. If an email address is not attached to your username, DigitalEdge will display an error message. Click **Manage Email Addresses** to associate an email address with your DigitalEdge username.


---


★ You can also assign an email address to your username by clicking your username in the upper right corner of the screen and selecting **User Settings**. Enter an address in the **Your Email Addresses** text box, click **Add**  and **Save** .

---

★ If other users are subscribed to this email notification, their email addresses will be listed below the **To** line. This is read-only information.

---

5. Enter a **Subject** line for this alert's email notification.
6. Compose the message body in the large text entry box.
7. Click **Create**  to save a new alert.

Or, click **Update**  to add an email definition to an existing alert filter.

The screenshot shows the DigitalEdge Alert Controller interface. On the left, a table lists alerts and their data models:

Name	Data Model
Hit	cef
baseball team	Alert

The main panel is titled 'Hit' and shows the 'Email Template' configuration for the 'cef' alert. The 'To' field is set to 'Me' with a 'Manage Email Addresses' link. The 'Subject' field contains the text: 'Alert generated for @distinctGameNumber@ on @dayOfWeek@'. The 'Body' field contains the text: 'On @gameDate@ the baseball team @winningTeam@ won on @dayOfWeek@'. A tip below the body field states: '[Tip:] This email can include dynamic data from your DigitalEdge system. To include a field from this alert's data model, simply surround a data model field name with @ symbols. For example'. Below the tip, an example is provided: 'Subject: New Preferred Customer - @user.username@' and 'Body: @user.firstName@ @user.familyName@ has become a preferred customer. Please email him/her congratulations at: @user.emailAddr@'. At the bottom, there are buttons for 'Undo My Changes', 'Update', and 'Cancel'.

As soon as you assign a username to an **Email Template**, the alert is activated; every time DigitalEdge matches an alert filter to new data, an email notification will be generated and sent out. Messages are sent individually; multiple alerts are not consolidated into one message.

You cannot delete an **Email Template**, but you can edit a template, blank out all its fields, and **Update** it so email notifications are no longer sent out for an **Alert Criteria**.

If you create and save an **Email Template** without assigning a username in the **To** field, alert notifications will be generated but not emailed. You might want to create such an alert message and write a user app that captures the message as a JMS message.



## Chapter 9: Creating Analytical Dashboards

Once your data is processed, enriched, indexed, and searchable, analysts and business intelligence administrators will want access to tools that analyze big data, interpret and extract information of interest, and present findings in an engaging display. One such tool is the **Pentaho® BI Suite Community Edition (CE)**, an open source tool for creating graphical and interactive dashboards. Pentaho can use DigitalEdge data as input to its suite of analytical tools, identify key indicators, transform the data into visual tools such as charts, tables, and maps, and tailor the output into a customized browser-based dashboard specific to your analytical needs. In a matter of seconds, raw data is transformed into a highly effective visual tool for immediate analysis and use.

Pentaho is just one tool that can potentially integrate with DigitalEdge. It is not bundled with DigitalEdge, but the open source version is freely available on the Internet should you choose to download it, learn it, and integrate its visual analytics with processed big data from DigitalEdge.

To create a dashboard with Pentaho, you should have the following tools:

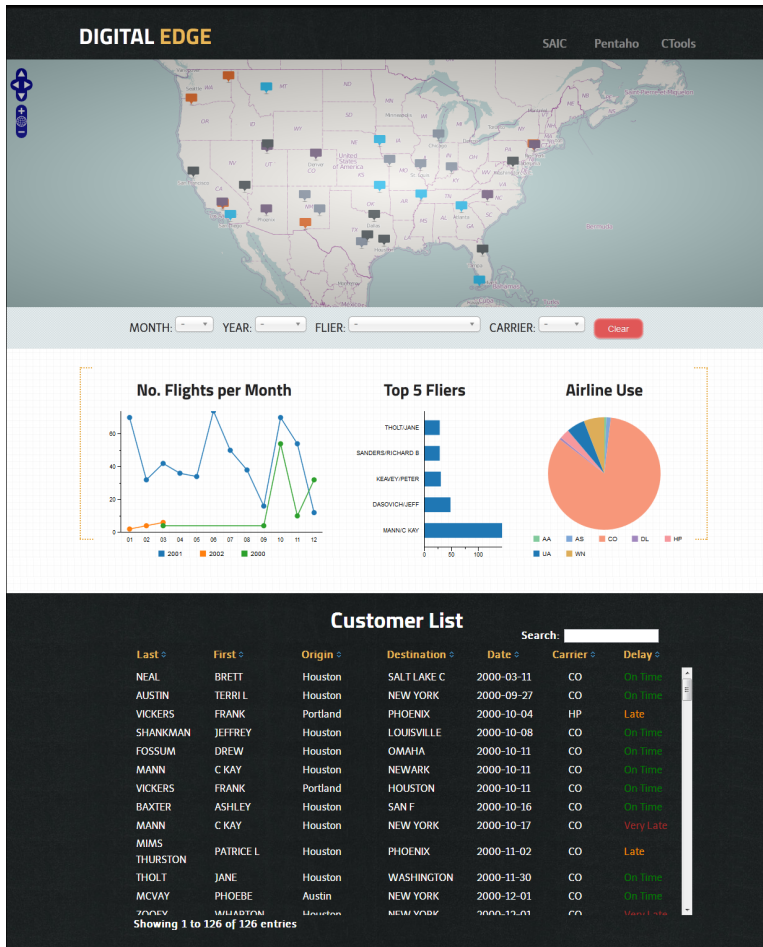
- The Pentaho BI Platform (Community Edition)
- Kettle: Pentaho's Data Integration (PDI) product, to define data transformations
- CTools: especially CDF, CBF, and CDA. CTools ("Community Tools") is an open source plug-in, developed by the Pentaho user community and managed by Webdetails.

Integrating Pentaho with DigitalEdge involves the following tasks:

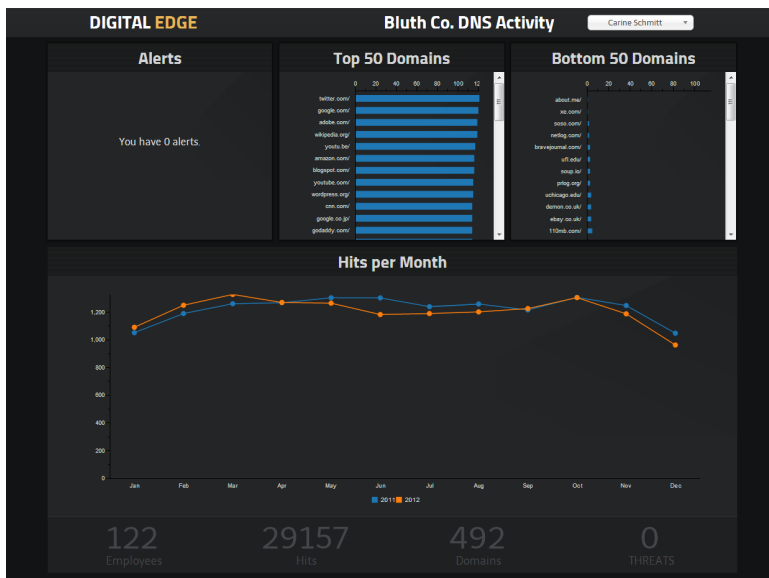
- Download and install the open source version of [Pentaho](#) and [CTools](#) from the Internet
- Learn how to use Pentaho and the CTools
- Configure a Pentaho webapp (contact Leidos for details)
- Prepare DigitalEdge data for input into a Pentaho dashboard. [See "Preparing data for dashboard input" on page 111.](#)
- Create a visual dashboard using Pentaho and its suite of configuration tools. [See "Creating a dashboard" on page 111.](#)
- Share the dashboard with analysts and business users. [See "Sharing a dashboard" on page 112.](#)

Dashboards are completely modular. You can select line, bar, or pie charts; data tables; or maps for inclusion. You can write scripts to interact with the data. And you can format and style the dashboard to your liking. Each dashboard is uniquely customized to your environment and needs.

For example, this dashboard was created by analyzing hundreds of thousands of email messages to cull flight and airfare information:



And this dashboard tracks DNS sites that employees have visited recently, alerting on sites that are potential sources of malware:



## Preparing data for dashboard input

To populate a dashboard, you need to share DigitalEdgeprocessed data by transporting it into a relational database (such as H2, Oracle, or PostgreSQL™) for Pentaho access. Then, use Pentaho's Kettle tool to define data transformations.

1. In Kettle, create a new transformation with the **Design** tab.
2. Select the **Big Data** tool to share normalized, enriched data from your DigitalEdge data sink (such as MongoDB) as the **Input**.
3. Using the **JSON Input** step, retrieve the fields that you will need for the dashboard.
4. Select **Table Output** to move data into a relational database that Pentaho can access.
5. Make sure that your Pentaho relational database includes columns that correspond to the transformation fields.
6. Click the **Preview the Transformation** icon and the **Quick Launch** button to test the transformation. Or, double-click on one step to **Preview** an individual step in the transformation.
7. To run the transformation, use the green **Play** button, or select **Action > Run**.

## Creating a dashboard

You can use the Pentaho Design Studio to create a dashboard, or the CDE plug-in component. CDE is just one component in the CTools plug-in, an open source suite of Pentaho tools which are developed and contributed by the Pentaho user community and managed by Webdetails. CTools ("Community Tools") go beyond Pentaho in the range of design features offered and the ease of use.

Steps for using CDE (Community Dashboard Editor) include:

1. Download and install [CTools](#).
2. Open the CDE plug-in to create a new Pentaho dashboard.
3. Create a dashboard with the following functions (in any order):
  - Use the **Layout** tab to format and style the dashboard components.
  - From the **Data Sources** tab, identify your data sources, make connections to the database, and submit queries.
  - From the **Components** tab, select the types of components you want to include in the dashboard (chart, tables, maps, etc.). Write an SQL query and run it against your data.

CDE creates a dashboard by linking these three elements together.

4. Optionally, style the dashboard with one of the component's advanced **Properties**, a Cascading Style Sheet (CSS), or a custom Javascript for interactivity.
5. Save the dashboard.

6. Analyze the CDA **Files** (Community Data Access), which list all your data sources and the results of the query, to determine if the query is providing the information you need. Refine the query as needed.
7. **Preview** the dashboard from CDE to make sure that the database connections and query are working as planned.

### Making interactive components

You can make components interactive by including parameters in your data source and adding listeners to your components. Add **Parameters** in CDE, then use the **Get Variables** function in Kettle to retrieve the parameters for the dashboard.

You can also make a component clickable by enabling the component's **Clickable** property and writing a script that defines the clickable change in the **Click Action** property.

### Adding maps

In CDE, select **Components/Custom/NewMapComponent** to add and configure an OpenStreet map.

### Sharing a dashboard

Once you have designed and tested a dashboard, you can share it with business analysts on your intranet.



You will need a Pentaho server webapp for DigitalEdgeSupport@Leidos.com before proceeding. Contact Leidos for assistance.

---

1. You can **Preview** the dashboard in CDE at any time.
2. To refresh a static dashboard, rerun the KTR transformation and **Preview** the dashboard again.
3. To share a dashboard on your intranet, first move a blank dashboard to a server by copying the CDA files from CDE. Then plug in the database connection addresses in the KTR transformation.



## Chapter 10: Indexing and Searching

Once you have a basic DigitalEdge system configured and running, you may want to index processed records for searching. DigitalEdge provides several features used to index data and to search records:

- **Lucene Indexing data sink:** You *must* add a Lucene data sink to the system to index DigitalEdge records. The Lucene data sink uses a copy of the processed DigitalEdge data and builds an inverted index for near real-time search.
- **Search API:** A REST API for near real time searches that helps you build a custom search interface for your system. The Solr™ search platform from Apache Lucene™ is embedded in the API to provide search services to the interface.
- **Search app:** A search application that provides a GUI as a quick way to search DigitalEdge records with minimal development.

Use the **System Builder** to configure new index and search components:

- **Configure an indexing data sink** to tokenize processed records and build an inverted index for a search feature; [See "Configuring an indexing data sink" on page 113](#)
- **Add webapps** for both the DigitalEdge Search API and the Solr™ search webapp; [See "Adding a search capability" on page 114](#)

You might also consider other specialized search webapps in DigitalEdge:


- **hue.server:** a web-based user interface for Apache Hadoop; requires the Hive data sink
- **phoenixapi:** a beta REST API for interacting with the Phoenix SQL query engine on HBase

### Configuring an indexing data sink

The first step in building a search capability is to create and configure the inverted index.

The Lucene data sink builds an inverted index that is optimized for near real-time data searching with the DigitalEdge search webapp. It stores index entries, not tully processed DigitalEdge records. The Lucene data sink is required for all search applications.

1. Open **System Builder**.
2. Go to the **Overview** tab.
3. Use the **Data Sinks** section.
4. Click **Add**.
5. Choose the **LuceneIndexingDataSink** from the **Select a Data Sink** dialog box and click **OK**.


6. Edit the default values in the **Set Datasink Parameters** dialog box. [See "Lucene indexing data sink" on page 77](#) for details about each parameter.
7. Click **OK**.
8. Click **Build** .

## Adding a search capability

You can add a search capability to your system at any time. There are several options to consider for a search capability:

- **Search app:** A search application provided with DigitalEdge. This app provides a quick way to search DigitalEdge records via a GUI with minimal development.
- **Search API:** A DigitalEdge REST API that helps you build a custom search capability for your system. The API uses the Solr™ search platform from Apache Lucene™.
- **Custom search client:** You can also build a custom search client with any web framework such as Apache Flex or HTML/CSS/Javascript.

To implement a full text search engine that provides a quick view of your data records, follow these steps:

1. Open **System Builder**.
2. Go to the **Overview** tab.
3. Use the **Webapps/REST APIs** section.
4. Click **Add**.
5. From the **Select a Webapp/REST API** dialog box, choose **search** for the Solr™ search capability.
6. From the **Select a Webapp/REST API** dialog box, choose the **searchapi**.
7. Click **OK**.
8. Click **Build** .



Consult the *Search API Guide* for details on customizing search features.



Contact an Leidos Support Engineer for details on configuring the search application to your site-specific needs.

---

## Using the search application

Once you build a Lucene indexing data sink and configure your processing pipeline to include both the **searchapi** and **search** webapps, you can access the DigitalEdge search interface to query your data.

Solr provides a rich set of search capabilities. The following features are implemented in the default DigitalEdge search interface:


- Full text searching
- Wildcard searching (\* for multiple characters; ? for one character)
- Boolean operators: AND, OR, and NOT (must be all caps)
- Phrase searching with quotation marks
- Nested queries in parentheses
- Stop word exclusion
- Expansion or collapse of full data records in search results
- Reset option to return to the full record set
- Index stemming

You can customize the search interface to take advantage of other Solr features, such as:

- Spelling suggestions
- Auto-complete search strings
- "More like this" searching for similar documents
- Faceted searching
- Highlighted words in context in search results
- Sorting by different fields
- Defining the title, summary, and body of records

To learn more about Solr's powerful feature set, consult the [Solr Wiki](#).

---

 Contact your Leidos Support Engineer assistance with configuring the search application to your specific needs.

---


To use the search interface:

1. Access the URL site that your DigitalEdge Administrator has established for your search application, such as:

`https://default.<system_domain_name>/search`

where `system_domain_name` is the full **Domain Name** created in **System Builder**.

---

 The search app uses the HTTPS protocol, not HTTP.

---

Provide your **Username** and **Password** to **Login**.

2. Enter a search in the **Search** text box at the top of the screen and press **Enter**. For example, when running the sales data model:

Example	To search for:
order.quantity_l:19	To search for a quantity of 19 in the order.quantity field
order.discount_d:0.01	To search for orders with an applied discount of 1% in the order.discount field
customer.customerName_s:Customer#000115915	To find orders from customer #000115915 in the customer.customerName field
ship.shipDate_dt:[2012-05-06T23:59:59Z TO NOW}	To find orders placed after May 6, 2012 in the ship.shipDate field

Note that each query must include a field type designation appended to the field name:

- \_l = long integer field type
- \_d = double numeric field type
- \_s = text string field type
- \_dt = date/time field type

Search fields will vary depending on the data model you are using; the online Search Help will list information about the search fields specific to your data model. For example, here are the searchable fields for the sales data model:

Data Model Field	Data Type	Search Field
content	text_rtws	content
customer.customerName	string	customer.customerName_s
customer.customerNation	string	customer.customerNation_s
line.lineItemComment	string	line.lineItemComment_s
line.lineNumber	long	line.lineNumber_l
line.lineStatus	string	line.lineStatus_s
order.clerk	string	order.clerk_s
order.discount	double	order.discount_d
order.discount	long	order.discount_l
order.extendedPrice	double	order.extendedPrice_d
order.orderComment	string	order.orderComment_s
order.orderDate	date/time	order.orderDate_dt
order.orderKey	string	order.orderKey_s
order.orderPrice	double	order.orderPrice_d

Data Model Field	Data Type	Search Field
order.orderPriority	string	order.orderPriority_s
order.orderStatus	string	order.orderStatus_s
order.quantity	long	order.quantity_l
order.tax	double	order.tax_d
part.partMfgr	string	part.partMfgr_s
part.partName	string	part.partName_s
part.partType	string	part.partType_s
ship.commitDate	date/time	ship.commitDate_dt
ship.receiptDate	date/time	ship.receiptDate_dt
ship.returnFlag	string	ship.returnFlag_s
ship.shipDate	date/time	ship.shipDate_dt
ship.shipInstructions	string	ship.shipInstructions_s
ship.shipMode	string	ship.shipMode_s
ship.shipPriority	string	ship.shipPriority_s
standardHeader.accessLabel	string	standardHeader.accessLabel_s
standardHeader.modelName	string	standardHeader.modelName_s
standardHeader.modelVersion	string	standardHeader.modelVersion_s
standardHeader.source	string	standardHeader.source_s
standardHeader.uuid	string	standardHeader.uuid
supplier.supplierName	string	supplier.supplierName_s
supplier.supplierNation	string	supplier.supplierNation_s

3. Search results are displayed with short citations. Search results tips:

- Click **more** to view a complete record for a specific citation.
- Click **less** to return to the short citation.
- Page through the results with the next > and previous < arrows.
- If your data includes date and time fields, click **Advanced Search** to access the **Time** search box. Click inside the **Time** search box to access a calendar and unit time search

boxes. Click **Done**.

- Click **Reset** to return to the full record set.

## Chapter 11: Troubleshooting

DigitalEdge offers several different tools for troubleshooting issues and problems. Check the health and status of the system with these tools:

Tool	Diagnostics
Management Console	Provides a quick snapshot of system health <a href="#">See "Viewing system status" on page 49</a>
System Monitor	Displays metrics about data processed within DigitalEdge <a href="#">See "Checking system metrics" on page 55</a>
Log files	DigitalEdge provides many log files for each process group and node. Log files record all activity and error messages for troubleshooting. The master node log file (master.log) contains much of the information associated with system initialization. <a href="#">See "Checking log files" on page 52</a>
Node documentation	Lists the commonly deployed DigitalEdge nodes/instances, their functions, and the services on each node <a href="#">See "What Each Node Does" on page 145</a>

This section also describes specific error conditions and solutions to some common problems. Consult these suggestions and tools before calling for support.

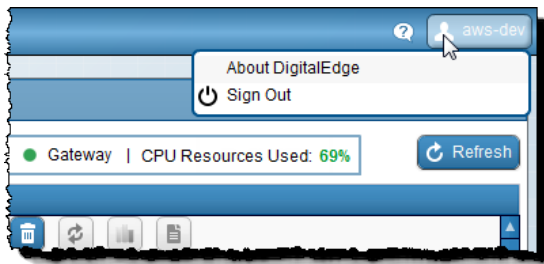
## Software version

### ? Issue

What version of DigitalEdge is running?

### ✓ Solution

To determine the DigitalEdge version you are running, on the **Management Console** or any of the setup/runtime tools, click the user icon in the upper right corner and select **About DigitalEdge**.





## Accessing applications



### Issue

How can I find out what web applications are available? How can I access one of them?



### Solution

Web apps that support a separate GUI, such as the **Alert Controller** and the **Search** application, can be accessed with a URL such as:

```
https://default.<system_domain_name>/<application_name>
```

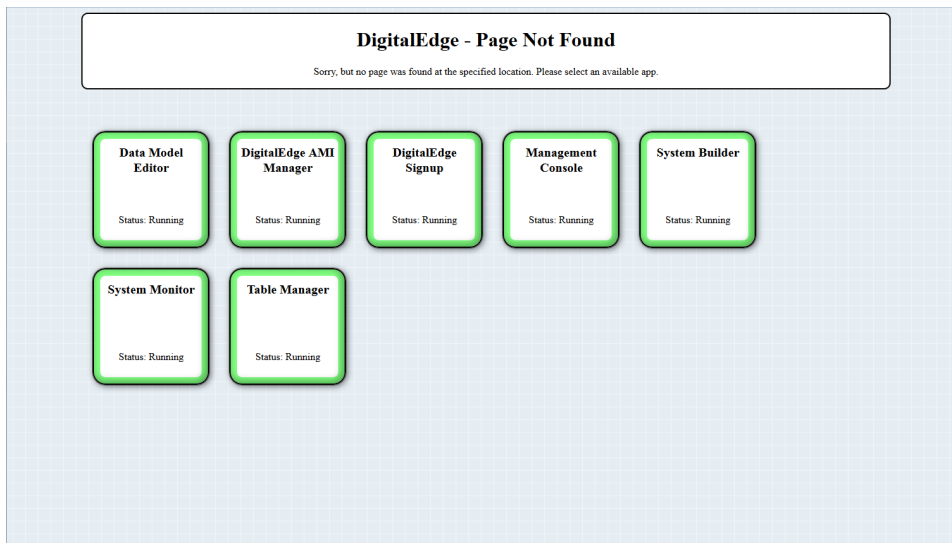
```
https://default.<system_domain_name>/search
```

where `system_domain_name` is the full **Domain Name** created in **System Builder**.

But if you do not know the application's name, or you don't know what web apps are available, you can access the router's splash page which redirects you to any running application:

```
https://default.<system_domain_name>/a
```

The splash page lists all applications and links you to the available apps. (Green = available, Red = not currently accessible). Click on any green box. For example:



## Data Transfer Utility: Connection error when sending data

### Issue

When using the Data Transfer Utility (DTU) application, a connection error occurs during an attempt to send data through the system.

### Solution

In the **Management Console > Security**, check your security groups. Most likely, a port is not open on the appropriate security group. The Data Transfer Utility uses port 61616. [See "Managing security groups and rules" on page 35](#) for instructions about opening a port.



## How to determine if a service is down

### Issue



How can I find out if a service is running or down?

### Solution

The **Management Console** displays warnings and messages when an instance fails. If a node is down, a warning will appear in the message box, such as:

 The system is reporting following errors:  
[jms.external]: Unable to connect to process group monitor for jms-ext-node1. 

or

 The system is reporting following errors:  
[hive.metastore]: [hive-metastore]: [HiveMetaStoreMonitor]: Hive Metastore server is currently unknown.  
[hive.metastore]: [hive-metastore]: [com.saic.rtw.common.monitor.process.H2Monitor]:  
'com.saic.rtw.common.monitor.process.H2Monitor' is an unknown process monitor.  
[datasink.hive]: [ingest-hive1]: [HiveServerMonitor]: Hive server status is currently unknown. 

The messages provide information at the node or instance level, not at the individual service level. To determine which services run on a node, [See "What Each Node Does" on page 145](#).

Double-click on the problematic Process **Group Name** in the **Management Console** to see a list of instances and IP addresses assigned to that node.

## How to troubleshoot data flow problems

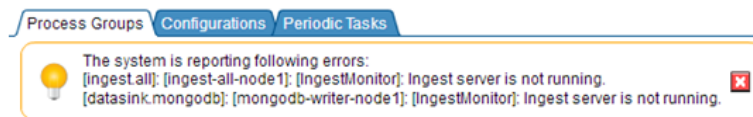
### ? Issue


Data is not flowing into/through my system as expected. How can I find out what's wrong?

### ✓ Solutions

1. Start with the **Management Console**.

If your system is in a **Warning** state, read the messages under the **Process Groups** tab. For example:

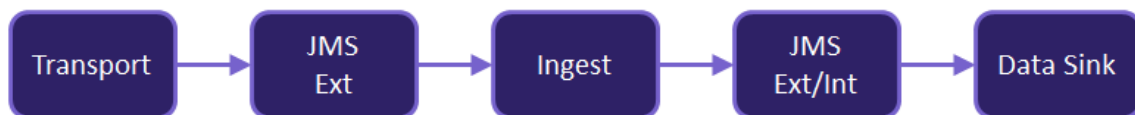


Use the **Log Viewer**  to examine the logs for the process groups identified in the warnings. For example:



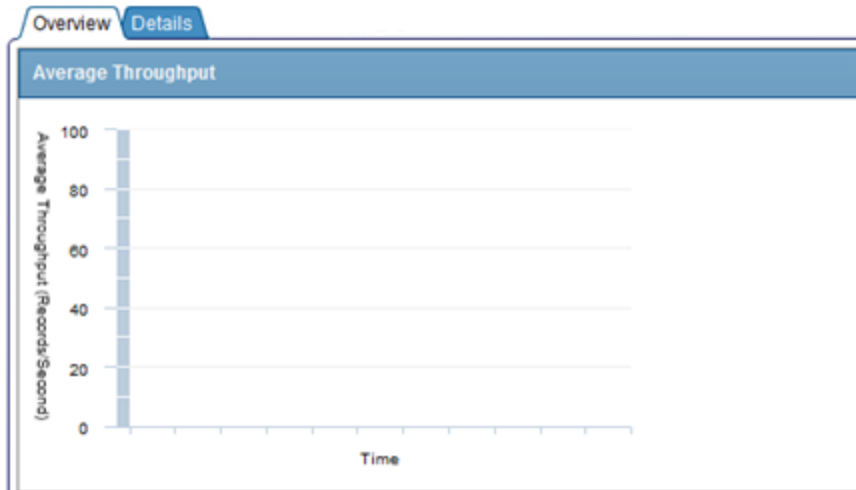
2. If your system status is **OK** but data is still not flowing, use the **System Monitor** and **Log Viewer** to diagnose the problem.


Data flows through DigitalEdge process groups in this order:



The problem could be in any one of these process groups.

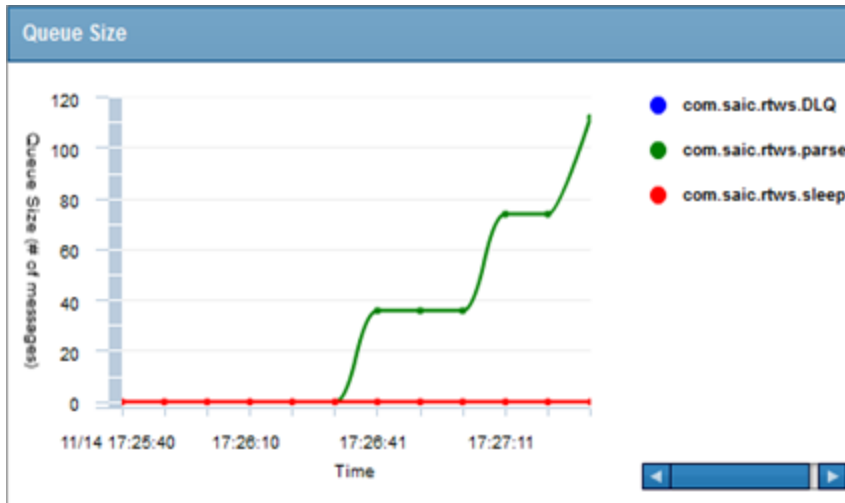
3. **Transport issue:** **System Monitor** shows no throughput:




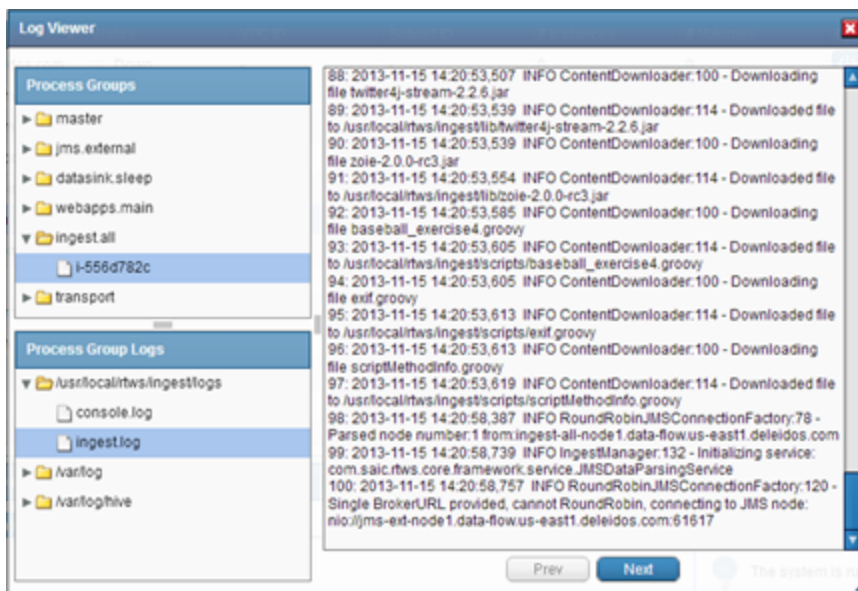
In the the **Log Viewer** , look under **transport** to examine `/usr/local/rtws/transport/logs-transport.log`:

The screenshot shows the Log Viewer interface. On the left, there are two panels: 'Process Groups' and 'Process Group Logs'. The 'Process Groups' panel shows a tree view with 'jms.internal', 'datasink.sleep', 'webapps.main', 'ingest.all', and 'transport'. The 'transport' group is selected, showing a sub-group 'i-7bcb5001'. The 'Process Group Logs' panel shows a list of logs for the 'transport' group, including 'console.log', 'transport.log', 'Avanlog', and 'Avanloghive'. The 'transport.log' is selected. On the right, the log content is displayed, showing a series of INFO messages from 'ContentDownloader' regarding file downloads. The logs include timestamps, process IDs, and file names. At the bottom, there are 'Prev' and 'Next' buttons.

- Ingest issue: System Monitor** shows no throughput for ingest.all but the parse queue is growing:




In the **Log Viewer** , look under **ingest.all** to examine `/usr/local/rtws/ingest/logs-ingest.log`:



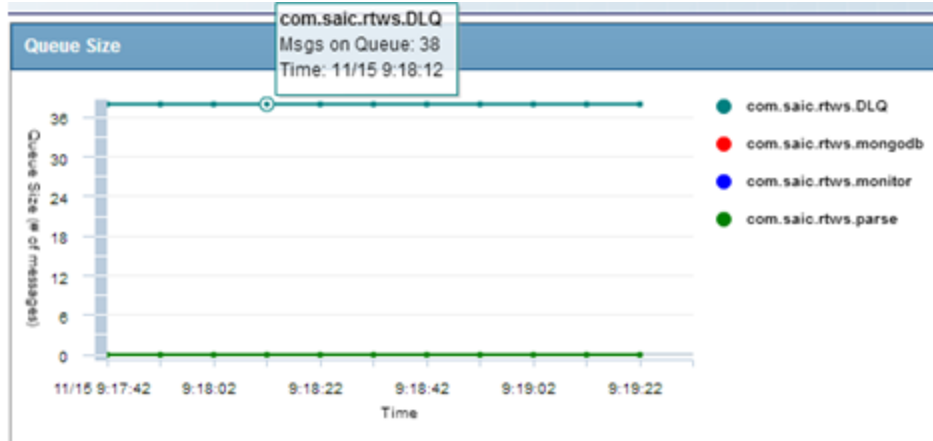
5. **Data Sink** issue: **System Monitor** shows throughput for ingest.all, no throughput for your data sink, but the data sink queue is growing:




In the **Log Viewer** , look under your **datasink** listing to examine `/usr/local/rtws/ingest/logs-ingest.log`.

- JMS** and other issues: If the previous steps did not identify the problem, check your JMS process groups. In the **Log Viewer** , look under **jms.external** and **jms.internal** to examine `/usr/local/apache-activemq/data/activemq.log`.

If your Dead Letter Queue (**DLQ**) is larger than 0, either ingest or the data sink is having trouble processing the received data:



You can view the contents of the **Dead Letter Queue** with the **Data Transfer Utility**.

Also, in the **Log Viewer** , look under both **ingest.all** and your **datasink** to examine `/usr/local/rtws/ingest/logs-ingest.log`.

## Ingest or transport process has run out of memory

### ? Issue

The transport or ingest process ran out of memory. How do I get it back to a working state?

### ✓ Solution

You can stop and restart a **transport**, **ingest.all**, or **datasink.lucene** process group without stopping the entire system.

In the **Management Console** on the **Process Groups** tab, double-click on the **Group Name** and **Restart** the problematic group. [See "Stopping and restarting a process group" on page 32](#) for details.

Contact DigitalEdge Technical Support to report the problem and to receive further guidance.

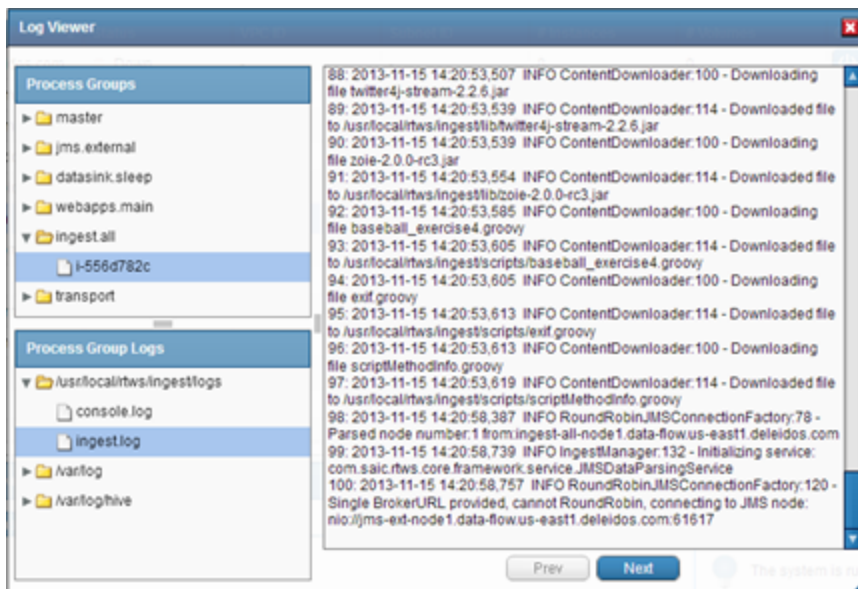
## Ingest.all is down

### ? Issue

The JMS parse queue is filling up, and the **Management Console** is warning that the ingest.all node is down.

### ✓ Solution

In the **Management Console**, use the **Log Viewer**  to examine the logs for the ingest.all node `/usr/local/rtws/ingest/logs-ingest.log`:



If the log indicates a minor problem, stop and restart the ingest process. [See "Stopping and restarting a process group" on page 32.](#)



## Ingested data is incorrect



### Issue

Data entering the system may be incorrect. How do I stop the system to correct the data model?



### Solution

1. In the **Management Console**, stop the transport and the ingest process so that data is not pulled off the JMS parse queue. [See "Stopping and restarting a process group" on page 32.](#)
2. Open up the JMS UI (ActiveMQ™) to inspect the data on the queue to determine if the data coming in is correct. Purge the data from the queue if the data is wrong.
3. Make any necessary changes to your data model in the **Data Model Editor**. Update the data model with the **Management Console**. [See "Updating a system" on page 33.](#)
4. Start the transport and ingest processes in the **Management Console**. [See "Stopping and restarting a process group" on page 32.](#)

## Lucene data sink is full



### Issue

The Lucene data sink is full. This is a *temporary* system, used to test indexing. I want to wipe out this index and start all over.

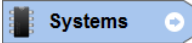


### Solution

To determine if Lucene is full:

1. In **System Monitor**, check the **Storage Utilization** chart. If there is no movement in the Lucene data sink line, indexing has stopped.
2. In the **Management Console**, check for a DigitalEdge warning message that indicates that the Lucene data sink is throttled. Click the **System** name you want to check on and look for the warning in the message box in the lower pane.

This solution to wipe out the Lucene index applies only to a test environment, not to a production system. If Lucene runs out of space in a production system, you must reconfigure your system and scale up. Contact DigitalEdge Technical Support for guidance.

1. In the **Management Console**, click the **Systems** option. 
2. Click the **System** name you want to work with.
3. On the **Process Groups** tab, double-click **datasink.lucene**.
4. Highlight the name of one or several process groups.
5. On the **Actions** menu, select **Purge Data** to delete the current index.


**Purge Data** will stop Jetty, wipe out the index data sink, and restart Jetty and the indexing process.

[See "Stopping and restarting a process group" on page 32](#) for more information.

## Management Console Gateway status is not green



### Issue

In the **Management Console**, the **Gateway** status is not  **OK**.



### Solution

See ["Viewing system status" on page 49](#) for details about the **Gateway** status display. If the **CPU Resources Used** stays in an **ERR** state, contact Leidos for assistance with the **Gateway** node.

## Management Console Gateway Resources Used are high

### ? Issue

In the **Management Console**, the **Gateway CPU Resources Used** status indicates a problem (an orange warning or a red error message), such as:

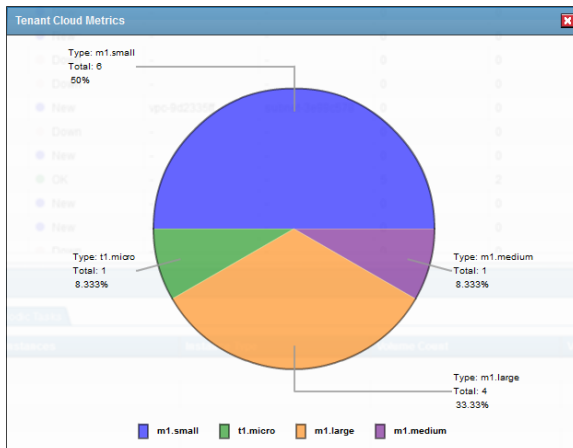
● Gateway | CPU Resources Used: **100%**

or

● Gateway | CPU Resources Used: **ERR**

### ✓ Solution

Click **CPU Resources Used** to view details about CPU resource utilization by the virtual machines in your tenant cloud environment:



To increase available CPU resources:

1. Shut down all unnecessary DigitalEdge systems running in **Management Console** (test systems, prototypes, outdated systems, etc.).
2. If available resources are still less than 25%, you should size up.
  - On an AWS cloud, increase your instance limits to allow for the creation of more instances. Call DigitalEdge Technical Support to reconfigure your system.
  - On a Eucalyptus-based system, add more hardware. Call DigitalEdge Technical Support for assistance.



To clear an error:

1. The Gateway status refreshes every 5 seconds; an Error status may resolve itself.
2. If the Gateway stays in ERR mode for several minutes, call DigitalEdge Technical Support.

## Management Console System Status = Warning



### Issue

In the **Management ConsoleSystems** page, a system displays a status of  **Warning** or  **Down**. A message appears in the **Management Console** such as “Unable to retrieve process group information for the system”, and no process groups appear in the **Group Name** column.



### Solution

Restart Jetty on the master node. If the status still doesn't change, contact Technical Support.

## Management Console System Status = Error



### Issue

In the **Management ConsoleSystems** page, a system displays a status of  **Error**.



### Solution

Log on to the master node and view the log files for that system ([See "Checking log files" on page 52](#)). If there are errors in a log file, or if the master daemon is not running, contact Technical Support.

## Management Console: Throttle condition message



### Issue

In a running DigitalEdge system, if a configured data sink has exceeded its storage capacity, the system will begin to drop subsequent records sent to the impaired data sink. The dropped records will not be recoverable by the system, and will not appear in any of the functionality supported by the affected data sink. This action is part of DigitalEdge's throttling system, to avoid loss of functionality provided by the data sink when one or more data sink can no longer process new data.

When this occurs, the **Management Console** displays a notification. A detailed message per data sink process group documents the throttle condition and a start date/time of the event, such as:

```
A throttle condition has triggered the dropping of messages. Start <Date  
Time>
```

During this time, **System Monitor** will no longer display processing metrics for the affected data sink. Once the affected data sink exits the throttling condition (if possible), the notification message is removed from the **Management Console**'s system process group information view, and the data sink will begin processing newly sent records.



### Solution

To avoid this throttling condition:

- Allocate a sufficient amount of storage for each data sink and their supporting process groups at system creation time.
- Allocate enough total instances per process group to satisfy the system's processing requirements.
- Develop a system policy to archive and/or purge data as required to avoid throttling in deployments where storage for system processing is limited.

## System Builder Error



### Issue

In **System Builder**, when you save or update a system configuration, an empty error dialog box appears without any error message.



### Solution

Most likely, the last component or parameter you were working with is missing a parameter value or includes an invalid parameter value. Update the parameter value(s) and save the system configuration again.

## System Monitor isn't working



### Issue

**System Monitor** graphs are blank and no data seems to be flowing in.



### Solutions

Make sure that the MetricsAPI on the webapps node of your system is up. If you just started up a system, the webapps node is one of the last nodes to come up.

If you have multiple DigitalEdge tabs open in your browser, check to see if any of those other applications have timed out. If so, close that tab and refresh the System Monitor page.

## System Monitor indicates scaling problems



### Issue

**System Monitor** indicates that **Queue Size** is very high and auto-scaling is working, but the queue is not being processed fast enough or the **Average Throughput** is not improving.



### Solution

On the newly added servers, check the ingest.all logs for errors ([See "Checking log files" on page 52](#)).

If ingest.all is processing, MetricsLog entries or errors should appear in the log. If that is not happening, check the process group logs for MetricsLog entries or errors. If you find any errors, contact Technical Support.

## System Monitor: Can't select a data model for Detail graphs



### Issue

On the **Details** graphs in **System Monitor**, the **Displaying Data Model** selection menu is blank.



### Solution

Close the **Displaying Data Model** menu and reopen it. If there are still no data models to select, restart Jetty on the webapps.main node and look for ingestAPI service errors.



## System Monitor Storage Utilization graph is blank



### Issue

The **Storage Utilization** graph in **System Monitor** displays no data, but the other **Overview** graphs are populated.



### Solution

Currently, the **Storage Utilization** graph only displays data for the Lucene data sink; the MetricsAPI webapp currently does not gather data from any other data sinks. If you configured a Lucene data sink in **System Builder** and believe that data should be appearing on the **Storage Utilization** graph, contact Technical Support.

## System Monitor, Management Console inconsistent number of instances



### Issue

In the **Management Console** **Systems** page, the **# Instances** do not match the number of instances reported in the **System Monitor's** **Number of Instances** graph.



### Solution

Restart Jetty on the master node and the webapps.main node.

## Tableau and the Hive data sink are not communicating



### Issue

When attempting to have the external Tableau® application access DigitalEdge-processed data in the Hive data sink, communication between Tableau and the Cloudera connection fails.



### Solution

To have Tableau® communicate with the DigitalEdge Hive data sink, follow these steps to establish the connection:

1. See the [Cloudera ODBC 1.2 Connector for Tableau](#) documentation for installation of the connector.
2. From Tableau Desktop, open port 1000 to the DigitalEdge server to support a Cloudera Hadoop Hive connection.
3. Specify authentication parameters in the Tableau Desktop configuration as follows:

Cloudera Hadoop Hive Connection

Step 1: Enter a server name:

< Hive\_IP\_Address > Port: 10000

Step 2: Select how to connect to the database

Type: HiveServer2

Authentication: User Name

Username: rtws

Realm:

Host FQDN:

Service Name:

Step 3: Establish the connection:

Connect

Step 4: Select a schema on the server:

default

Step 5: Select a table or view from the database:

☒ Single Table ☐ Multiple Tables ☐ Custom SQL

Step 6: Give the connection a name for use in Tableau:

Initial SQL... OK Cancel



The Hive IP address is specified in DigitalEdge at: **Management Console > Systems > Selected System > Process Groups > datasink.hive > Public IP**. Highlight the **datasink.hive** line and use the **Copy** command to put the IP address on the clipboard for pasting into the Tableau Desktop screen.

---



## Appendix A: Terminology

Term	Definition
Alert	Email notification to a user that a potential fraud has been detected by DigitalEdge
AMI (Amazon Machine Image)	A bootable server that is a special type of pre-configured virtual machine in the cloud; AMIs serve as basic units of service deployment
Anchor node	In TMS, the anchor node hosts CAS, LDAP, and the TMS database
AWS™ (Amazon Web Services™)	Remote web services that comprise the cloud computing platform offered by Amazon and implemented in the Eucalyptus platform
Data sink	A queue, server, or database that can receive pipeline-processed JSON data to store or post-process for other uses
Dead letter queue	If an incoming record cannot be parsed for any reason, rather than ignoring it and dropping it out of the system, DigitalEdge saves the record in the dead letter queue where you can examine it and correct it.
EC2™ (Amazon Elastic Compute Cloud™)	A key part of Amazon's AWS™ public cloud computing platform, providing users the ability to create, launch, and end virtual server instances in a scalable deployment of applications
Elastic IP	A static IP address designed for cloud computing, associated with your Amazon account, not an instance; EC2 lets you mask problems by remapping an elastic IP address to a replacement instance
EMI (Eucalyptus Machine Image)	A pre-configured virtual machine, including the operating system and virtual application software, that can be used to create an instance in a Eucalyptus environment
Gateway node	A node in a tenant system that hosts CAS for single sign on permissions and LDAP for user account credentials. The Gateway node starts and stops systems, creates and deletes systems and security groups, and synchronizes components.
Hybridfox	An optional Firefox add-on that provides an interface to cloud accounts, including AWS and Eucalyptus, to help you manage images, instances, security groups, key pairs, elastic IPs, and storage.

Term	Definition
IAAS	The Infrastructure As a Service (IAAS) number is a tenant ID assigned by the facility providing your cloud services (e.g., Amazon or Eucalyptus). In the Eucalyptus Console, it is identified under <b>Identity Management &gt; Accounts &gt; ID#</b> . It is always a 12 digit number. In the DigitalEdge Management Console, it is listed as the account ID.
Master node	A VM (virtual machine) that launches all other nodes in a system. The master node handles auto-scaling, internal monitoring, starting and stopping for all instances. In TMS, the master node includes the Master Repository. In a tenant account, the master node includes the System Repository.
Master Repository	The Master Repository resides in TMS. It is the storage location for all common plug-in components provided with DigitalEdge, and private plug-in components used by each tenant account.
NAT (Network Address Translation)	An instance which is configured to perform network address translation and to serve as a firewall into the private Amazon VPC subnet
POC	Point of Contact information
Private IP	An internal RFC 1918 address that is only routable within the EC2 Cloud; traffic outside your EC2 network cannot access this IP
Public IP	An Internet routable IP address assigned by the system for all instances; Traffic routed to a public IP is translated via NAT and forwarded to an instance's private IP address
Repository	The storage location for all the plug-in components; the System Repository stores private components used in a tenant's DigitalEdge account, the Master Repository resides at the TMS level and stores all common and private components
S3™ (Amazon Simple Storage Service™)	The online storage web service provided with AWS™ and used as a data source for public cloud instantiations
Splitter	Each transport works with a specific incoming record type (JSON, XML, PCAP, etc.); the transport's record-format parameter uses a splitter to define record boundaries when the input data includes multiple records
Tenant account	A tenant is an account on a cloud platform. In the public cloud, a tenant account typically represents an organization that is building an AWS application. On a private cloud,

Term	Definition
Primary tenant	<p>internal to an organization, a tenant account is usually a project or a department that runs its own secure applications.</p> <p>A <i>primary</i> tenant is the first tenant created in a DigitalEdge account (via the Installation program on a Eucalyptus system, via Registration on AWS systems). The primary tenant owns all the DigitalEdge resources: the system repository, LDAP, the tenant database, etc. and does not share data with other tenants.</p>
Secondary tenant	<p>One or more <i>secondary</i> tenants may be created in an account. A secondary tenant is created by a TMS Administrator in the Management Console. All secondary tenants share the account resources that are owned by their primary tenant (system repository, LDAP, etc.), share and see all systems created under an account, and have the same privileges as the primary tenant. But secondary tenants have different logon credentials for security purposes.</p>
TMS (Tenant Management System)	<p>The Tenant Management System is a behind-the-scenes infrastructure for DigitalEdge to create and manage tenant accounts. TMS provides services to create new accounts, to monitor tenant applications, to calculate tenant usage activity and charges, to manage user identities and permissions, to manage the DigitalEdge GUI tools and plug-in components, and to provide security.</p>
VPC™ (Amazon Virtual Private Cloud™)	<p>An isolated environment within the AWS cloud where you can launch applications in a more secure, virtual network</p>





## Appendix B: What Each Node Does

Each node in DigitalEdge is a virtual machine and an instance of a process group, most of which are auto-scaling. To help size a system, the following table provides details about what each node does.

Node	Content
webapps.main (on TMS)	Home to all the DigitalEdge APIs, Setup tools, and Runtime tools, including: <ul style="list-style-type: none"> <li>• Management Console</li> <li>• Data Modeler</li> <li>• Table Manager</li> <li>• System Builder</li> <li>• System Monitor</li> </ul>
anchor (on TMS)	Security and authentication node, housing: <ul style="list-style-type: none"> <li>• CAS (JA-SIG Central Authentication Service)</li> <li>• LDAP</li> <li>• TMS database</li> <li>• TMS Gateway</li> </ul>
gateway	The node that controls a DigitalEdge system, including: <ul style="list-style-type: none"> <li>• Launching the master node</li> <li>• Starting and stopping systems</li> <li>• Creating and deleting systems and security groups</li> <li>• Synchronizing components and repositories</li> <li>• Housing: <ul style="list-style-type: none"> <li>◦ CAS for single sign-on permissions</li> <li>◦ LDAP for user account credentials</li> <li>◦ APIs</li> <li>◦ Tenant database</li> </ul> </li> </ul>
master	The management node of DigitalEdge, controlling: <ul style="list-style-type: none"> <li>• Starting and stopping all instances</li> <li>• Monitoring for auto-scaling</li> <li>• Adding and removing nodes based on load and storage utilization</li> <li>• Handling virtual storage allocations</li> </ul>

Node	Content
	<ul style="list-style-type: none"> <li>• Gathering metrics for auto-scaling decisions</li> <li>• Housing the System Repository</li> </ul>
transport	Controlling all transports through the Transport API
jms.external	<p>First entry point into DigitalEdge, and a staging area for incoming data to:</p> <ul style="list-style-type: none"> <li>• Receive data pushed into the jms.external queue by other clients</li> <li>• Feed data directly into DigitalEdge</li> <li>• Manage the parsing queue</li> <li>• Receive processed alerts from the datasink.alert that match alerting criteria, and place a message in this queue for notifications</li> </ul>
ingest.all	<p>Ingest node to handle processing pipeline tasks, including:</p> <ul style="list-style-type: none"> <li>• Parsing</li> <li>• Enrichment</li> </ul>
jms.internal	<p>Internal staging area for the next steps in the processing pipeline; a buffer for records queued up waiting for the next phase of processing:</p> <ul style="list-style-type: none"> <li>• Post-enrichment record holding</li> <li>• Temporary record storage</li> </ul>
datasink	<p>Each type of data sink has its own node and processes data for specialized uses; for example:</p> <ul style="list-style-type: none"> <li>• datasink.alert - filtering records against alert criteria, sending alert messages to the configured recipient (such as a topic on the jms.external node, an email message, etc.)</li> <li>• datasink.hbase - storing records to the Hadoop Distributed File System (HDFS)</li> <li>• datasink.hive - storing records to HDFS</li> <li>• datasink.lucene - indexing records for searching</li> <li>• datasink.mongodb - storing JSON-based records and providing a query interface</li> </ul> <p>Some data sinks automatically add additional nodes when they are spawned; for example, HBase and Hive add nodes (such as zookeeper) that are needed for a complete HBase ecosystem</p>

Node	Content
webapps.main (on tenant)	<p>Home to all webapps and REST APIs, including:</p> <ul style="list-style-type: none"><li>• Search app</li><li>• Metrics API</li></ul>

# Index

## A

account ID [142](#)

Alert Controller

building [102](#)

how to use [105](#)

alerting data sink [102](#)

building [102](#)

parameters [66](#)

alerting system

building [102](#)

alerting web app [102](#)

building [102](#)

alerts

building [102](#)

business rules [101-102](#)

configuring [101-102](#), [105](#)

criteria [101-102](#)

data sink [102](#)

defined [141](#)

email subscriptions [101](#), [105](#)

filters [101-102](#)

notifications [101](#), [105](#)

setting up [101](#)

triggers [101-102](#)

Alerts API

building [102](#)

allocating resources

processor groups [82](#)

Amazon EC2

defined [141](#)

Amazon Elastic Compute Cloud

defined [141](#)

Amazon Machine Image

defined [141](#)

Amazon S3

defined [142](#)

Amazon Simple Storage Service

defined [142](#)

Amazon Virtual Private Cloud

defined [143](#)

Amazon VPC

defined [143](#)

Amazon Web Services

defined [141](#)

security [7](#)

AMI

defined [141](#)

analytics

dashboards [109](#)

anchor node

defined [141](#)

described [49](#)

IP address [49](#)

applications

accessing [121](#)

architecture

high level [3](#)

autoscaling

configuring [81-82](#)

troubleshooting [136](#)

AWS

defined [141](#)

security [7](#)

## B

building a system

nodes [145](#)

## C

CAS [7](#)

Cassandra data sink

parameters [68](#)

CDE [111](#)

Central Authentication Service [7](#)

certificates [7](#)

downloading [37](#)

uploading [37](#)

components

deleting [43](#)

managing [39](#)

uploading [41](#)

configuration

nodes [145](#)

configurations

deleting [35](#)

configuring

autoscaling [82](#)

process groups [82](#)

scaling [82](#)

users [38](#)

CSV parser

metrics [24](#)

CTools [109](#), [111](#)

downloading [109](#)

## D

dashboards [109](#)

creating [111](#)

CTools [111](#)

preparing data [111](#)

publishing [112](#)

sharing [112](#)

viewing [112](#)

data

ingesting [11](#), [17](#)

security [7](#)

data flow

monitoring [57](#)

troubleshooting [123](#)

data labeling [7](#)

data markings [7](#)

## data models

- associating with data sinks [65](#)
- missing in System Monitor [136](#)
- updating [33](#)

## data sinks

- adding [65](#)
- associating with data models [65](#)
- defined [141](#)
- dropping records [134](#)
- editing [84](#)
- exceeding storage capacity [134](#)
- indexing [113](#)
- Lucene [113](#)
- monitoring problems [137](#)
- parameters [66](#), [84](#)
- ScriptingDataSink [80](#)
- sizing [134](#)
- throttling [134](#)
- troubleshooting [125](#)

Data Transfer Utility [11](#)

- connection error [122](#)
- using [17](#)

database watcher transport [12](#)

- parameters [86](#)

## dead letter queue

- defined [141](#)
- troubleshooting [126](#)
- viewing [20](#), [55](#)

definitions [141](#)

## deleting

- configurations [35](#)
- system configurations [35](#)
- systems [31](#), [34](#)
- users [39](#)

detail graphs [60](#)

## DigitalEdge

- version [120](#)

## dimension

- data sink parameters [68](#)

directory crawler transport [12](#)

- parameters [88](#)

directory watcher transport [12](#)

- parameters [89](#)

## DLQ

- troubleshooting [126](#)
- viewing [20](#), [55](#)

## documentation

- types [1](#)
- typographical conventions [1](#)

DTU [17](#)**E**

## EC2

- defined [141](#)

## Elastic Compute Cloud

- defined [141](#)

elastic IP

defined [141](#)

elasticsearch data sink

parameters [69](#)

email alerts [105](#)

EMI

defined [141](#)

error status

Management Console [133](#)

Eucalyptus

security [7](#)

Eucalyptus Machine Image

defined [141](#)

external apps

linking to DigitalEdge [80](#)

external HBase data sink

parameters [69](#)

external Hive data sink

parameters [70-71](#)

## F

Federal Information Security Management Act [7](#)

fine-tuning a system [65](#)

FISMA [7](#)

## G

gateway node

defined [141](#)

described [6](#)

IP address [49](#)

problem [131](#)

resources low [132](#)

resources used [49](#)

status [49](#)

getting data in [11](#)

Data Transfer Utility [17](#)

unstructured [15](#)

glossary [141](#)

graphs

Average Throughput [57](#)

Data Model Processing Time [62](#)

detail [60](#)

Ingest Processing Time [62](#)

Number of Instances [59](#)

overview [56](#)

Processor Processing Time [63](#)

Queue Size [58](#)

Storage Utilization [60](#)

System Monitor [57](#)

## H

HBase

parameters [73](#)

Hive data sink

parameters [75](#)

troubleshooting [138](#)

Hive transport [12](#)

parameters [91](#)

Hue [113](#)

Hybridfox

defined [141](#)

## I

IAAS number

defined [142](#)

indexing [113](#)

configuring a data sink [113](#)

indexing data sinks

parameters [77](#)

Infrastructure As a Service number

defined [142](#)

ingest

troubleshooting [124](#), [127-129](#)

ingest times

monitoring [62](#)

input models

updating [33](#)

instances [145](#)

monitoring [59](#)

problems [137](#)

troubleshooting [137](#)

IP addresses

Amazon [84](#)

assigning [84](#)

Eucalyptus [83](#), [85](#)

gateway [49](#), [84](#)

viewing [49](#), [51](#), [84](#)

VPC [85](#)

webapps.main [83](#), [85](#)

IP, elastic

defined [141](#)

IP, private

defined [142](#)

IP, public

defined [142](#)

## J

JA SIG Central Authentication Service [7](#)

JMS

troubleshooting [126](#)

JMS bridge transport [12](#)

parameters [92](#)

jobs

about [44](#)

creating [45](#)

deleting [46](#)

scheduling [45](#)

JSON to JDBC data sink

parameters [76](#)

## K

Kettle [109](#)

data transformations [111](#)

keystores

downloading [37](#)

uploading [37](#)



**L**

## log files

accessing [52](#)described [52](#)security [8](#)troubleshooting [54](#)types [52](#)logging in [30](#)logging out [46](#)Lucene [113](#)configuring [113](#)parameters [77](#)troubleshooting [130](#)**M**

## Management Console

adding a user [38](#)dashboard [29](#)deleting a configuration [35](#)deleting a system [34](#)deleting a system configuration [35](#)error status [133](#)functionality [29](#)gateway resources low [132](#)gateway status problem [131](#)how to use [29](#), [49](#)instances problem [137](#)log files [52](#)opening a port [84](#)plug-ins [39](#), [41](#), [43](#)restarting a process group [32](#)restarting a system [31](#)security [8](#)security features [8](#)security groups [36](#)security rules [36](#)service is down [122](#)starting a system [29-30](#)status codes [50](#)stopping a process group [32](#)stopping a system [29](#), [31](#)system status [29](#), [49](#)throttle condition [134](#)updating a data model [33](#)uses [29](#)warning status [133](#)

## master node

defined [142](#)described [6](#)

## Master Repository

defined [142](#)uploading to [39](#)

## metrics

CSV files [24](#)systems [55](#)

## MongoDB data sink

parameters [79](#)

MongoDB transport [12](#)

parameters [93](#)

monitoring

data flow [57](#)

ingest times [62](#)

instances [59](#)

processing rates [58](#)

processing times [62](#)

queue sizes [58](#)

resource scaling [59](#)

storage [60](#)

throughput [57](#)

## N

NAT

defined [142](#)

network address translation

defined [142](#)

nodes [145](#)

notifications

setting up [101](#)

## O

opening a port [84](#)

overview graphs [56](#)

## P

passwords

changing [47](#)

pcap transports [12](#)

parameters [94](#)

PcapSnifferTransportService [12](#)

parameters [94](#)

Pentaho

dashboards [109](#)

data transformations [111](#)

downloading [109](#)

Pentaho BI Platform [109](#)

periodic tasks

about [44](#)

creating [45](#)

deleting [46](#)

scheduling [45](#)

Phoenix

web app [113](#)

plug-in components

managing [39](#)

plug-ins

deleting [43](#)

uploading [41](#)

POC

defined [142](#)

PollingS3FileTransportService

example [14](#)

primary tenants

defined [142](#)

private components

deleting [43](#)

managing [39](#)

- uploading [41](#)
- private IP
  - defined [142](#)
- process groups
  - configuring [81-82](#)
  - described [82](#)
  - parameters [81-82](#)
  - restarting [32](#)
  - security groups [35](#)
  - stopping [32](#)
- processing rates
  - monitoring [58](#)
- processing times
  - monitoring [62](#)
- public IP
  - defined [142](#)
- Q**
- queue sizes
  - monitoring [58](#)
- R**
- repositories [39](#)
  - defined [142](#)
  - managing [39](#)
  - Master defined [142](#)
  - Master Repository, uploading [39](#)
- resource scaling
  - monitoring [59](#)
- resource utilization [49](#)
- resources used
  - gateway node [132](#)
- restarting
  - process groups [32](#)
- running a system [30](#)
- S**
- S3
  - defined [142](#)
- S3 file transports
  - example [14](#)
- S3FileTransportService [13](#)
  - example [14](#)
  - parameters [95](#)
- scaling
  - configuring [82](#)
  - troubleshooting [136](#)
- scheduled jobs
  - about [44](#)
  - creating [45](#)
  - deleting [46](#)
- ScriptingDataSink
  - configuring [80](#)
- Search API [113](#)
  - configuring [114](#)
- Search app [113](#)
  - configuring [114](#)
  - using [114](#)

- searching [113](#)
  - configuring a data sink [113](#)
  - configuring a webapp [114](#)
  - Hue [113](#)
  - Phoenix [113](#)
  - using the webapp [114](#)
- secondary tenants
  - defined [142](#)
- security [7](#)
  - architecture [7](#)
  - certificates [7](#)
  - components [7](#)
  - data [7](#)
  - data labeling [7](#)
  - data markings [7](#)
  - firewalls [8](#)
  - log files [8](#)
  - Management Console communications [8](#)
  - Management Console features [8](#)
  - passwords [47](#)
  - perimeters [8](#)
  - user auditing [8](#)
  - user authentication [8](#)
  - user management [8](#)
- security groups
  - described [35](#)
  - security rules [36](#)
- security rules
  - adding [36](#), [84](#)
  - deleting [37](#)
- Simple Storage Service
  - defined [142](#)
- sizing
  - processor groups [82](#)
- sleep data sink
  - parameters [80](#)
- software version [120](#)
- Solr [113](#)
  - using [114](#)
- splitters
  - defined [142](#)
- standards
  - documentation [1](#)
- starting a system [30](#)
- statistics
  - CSV files [24](#)
- status
  - error [133](#)
  - gateway node [131](#)
  - systems [51](#)
  - warning [133](#)
- status codes
  - described [50](#)
- stopping
  - process groups [32](#)

- systems [31](#)
- storage
  - monitoring [60](#)
- Storage Utilization graph
  - blank [137](#)
- style conventions
  - documentation [1](#)
- subscriptions
  - setting up [101](#)
- system architecture
  - high level [3](#)
- System Builder
  - adding a data sink [65](#)
  - adding a user app [65](#)
  - autoscaling [81](#)
  - configuring external apps [80](#)
  - configuring search [113-114](#)
  - error [135](#)
  - reallocating resources [82](#)
  - resizing a processor group [82](#)
  - search [113](#)
  - system changes [65](#)
  - transports [11](#)
- system configurations
  - deleting [35](#)
- System Monitor
  - Average Throughput [57](#)
  - blank graph [137](#)
- controls [56](#)
- Data Model Processing Time [62](#)
- data models missing [136](#)
- detail graphs [60](#)
- graphs [57](#)
- how to use [55](#)
- Ingest Processing Time [62](#)
- instances problem [137](#)
- logging out [63](#)
- not working [135](#)
- Number of Instances [59](#)
- overview graphs [56](#)
- Processor Processing Time [63](#)
- Queue Size [58](#)
- scaling problems [136](#)
- settings [56](#)
- Storage Utilization [60](#)
- system monitoring [49](#)
  - Management Console [49](#)
  - metrics [55](#)
  - status [49](#)
  - tools [49](#)
- System Repository
  - defined [142](#)
- system status
  - Management Console [51](#)
- system status codes
  - described [50](#)

## systems

- changing [65](#)
- deleting [31](#), [34](#)
- fine-tuning [65](#)
- launching [30](#)
- restarting [31](#)
- running [30](#)
- starting [30](#)
- status [49](#), [51](#)
- stopping [31](#)

## T

## Tableau

- troubleshooting [138](#)

## tasks

- about [44](#)
- creating [45](#)
- deleting [46](#)
- scheduling [45](#)

TCP transport [13](#)

- parameters [96](#)

## tenant account

- defined [142](#)

tenant ID [142](#)

## Tenant Management System

- defined [143](#)

## tenants

- defined [142](#)
- primary [142](#)

- secondard [142](#)

- terminology [141](#)

- throttle condition [134](#)

## throughput

- monitoring [57](#)

## TMS

- defined [143](#)

## transports

- custom [11](#)

- Data Transfer Utility [17](#)

- database watcher [12](#)

- directory crawler [12](#)

- directory watcher [12](#)

- editing [84](#)

- examples [13](#)

- getting data in [11](#)

- Hive [12](#)

- JMS bridge [12](#)

- MongoDB [12](#)

- parameters [84](#), [86](#)

- pcap sniffer [12](#)

- planning [11](#)

- TCP [13](#)

- troubleshooting [123](#), [127](#)

- Twitter filter [13](#)

- Twitter REST [13](#)

- Twitter sample [13](#)

- UDP [13](#)

- URL [13](#)
- triggers
  - alerts [101](#)
- troubleshooting [119](#)
  - accessing applications [121](#)
  - connection error [122](#)
  - data flow [123](#)
  - data sinks [125](#)
  - data sinks dropping records [134](#)
  - data sinks throttling [134](#)
  - Data Transfer Utility [122](#)
  - dead letter queue [126](#)
  - DLQ [126](#)
  - DTU [122](#)
  - gateway node status [131](#)
  - gateway resources used [132](#)
  - Hive data sink [138](#)
  - ingest [124](#), [127-129](#)
  - JMS [126](#)
  - log files [54](#)
  - Lucene [130](#)
  - Management Console error status [133](#)
  - Management Console warning status [133](#)
  - missing data models [136](#)
  - number of instances [137](#)
  - service is down [122](#)
  - software version [120](#)
  - Storage Utilization graph [137](#)
  - System Builder error [135](#)
  - System Monitor not working [135](#)
  - Tableau [138](#)
  - throttle condition [134](#)
  - tools [119](#)
  - transports [123](#), [127](#)
  - web apps [121](#)
- truststores
  - downloading [37](#)
  - uploading [37](#)
- Twitter filter transport [13](#)
  - parameters [96](#)
- Twitter REST transport [13](#)
  - parameters [97](#)
- Twitter sample transport [13](#)
  - parameters [98](#)
- typographical conventions
  - documentation [1](#)
- U**
- unprocessed records
  - reading [55](#)
- unstructured data
  - getting it in [15](#)
- UPD transport [13](#)
- URL transport [13](#)
  - parameters [100](#)
- user apps
  - adding [65](#)

users

- adding [38](#)
- auditing [8](#)
- authentication [8](#)
- configuring [38](#)
- deleting [39](#)
- editing [39](#)
- managing [8](#)

**V**

version [120](#)

Virtual Private Cloud

- defined [143](#)

VPC

- defined [143](#)

**W**

warning status

- Management Console [133](#)

web apps

- accessing [121](#)
- Hue [113](#)
- Phoenix [113](#)

webapps.main

- IP address [85](#)