DigitalEdge™

Overview Guide

Version 1.2.1

July 2014

leidos

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 Incorporate 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.

CONTACT INFORMATION

Leidos Franklin Center
6841 Benjamin Franklin Drive
Columbia, Maryland 21046

Email: 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

Introduction

The twenty-first century is experiencing an explosion of data; enormous streams of data are created every minute. The companies we work for, the electronic devices we use (cell phones, tablets), the organizations we interact with (banks, online stores, healthcare services, police departments), transportation modes (cars, airlines, rail), and sensors in infrastructures (bridges, oil rigs) all generate large data streams that are difficult to deal with. In fact, the world's information doubles approximately every two years. This massive amount of information is referred to as "big data". Challenges include capturing and storing data, searching and analyzing up-to-the-minute data quickly, and making timely, informed decisions based on the data. With 80% of data being unstructured or semi-structured, very large data sets cannot be dealt with using traditional tools such as relational databases, desktop software, and traditional servers. And with data volumes increasing exponentially, there is a pressing need for new systems to handle data in unprecedented amounts of terabytes, petabytes, and exabytes; not just to store it, but to process the data, analyze the data, and to use the data for informed decision making in a timely manner.

DigitalEdge Platform

DigitalEdge is an advanced, customizable data integration platform that enables a full lifecycle from data discovery to actionable intelligence. DigitalEdge integrates ETL (extract, transform, and load) real-time stream processing, and big data NoSQL ("Not Only SQL") stores into a high performance analytic system. Instead of waiting hours for actionable data and reports, analysts can now achieve near real-time situational awareness. DigitalEdge provides a rapid response to changing environments.

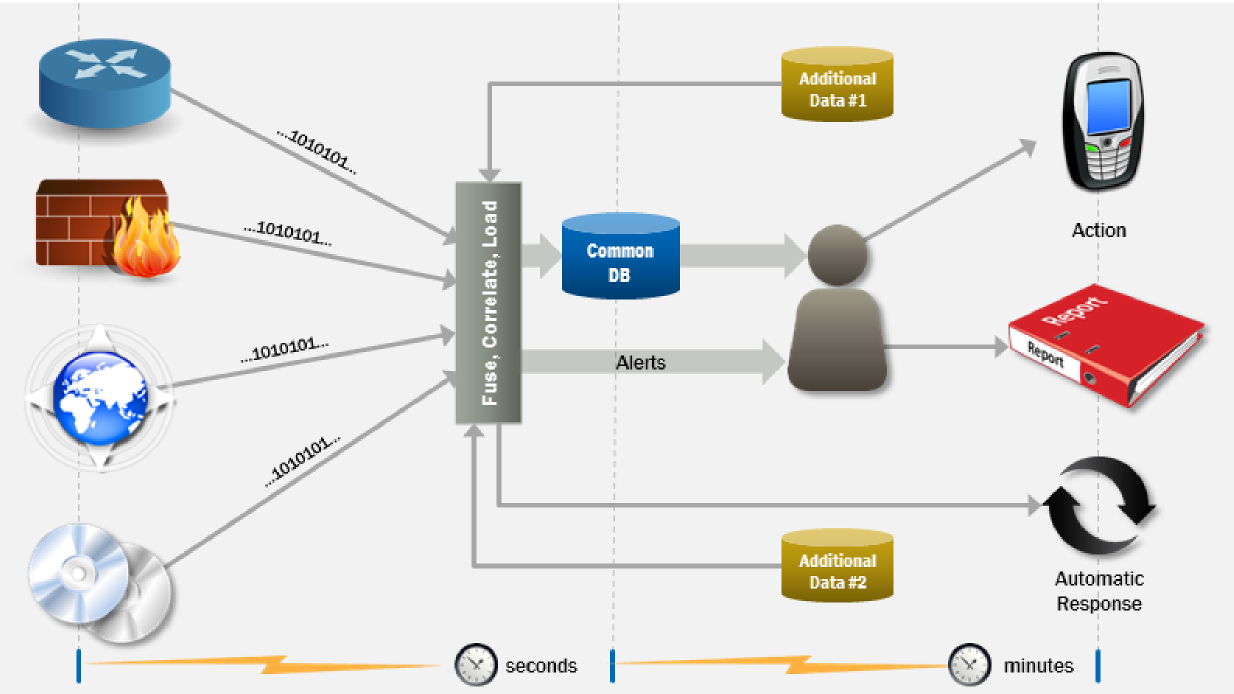
DigitalEdge is Java based and provides an extensible architecture, APIs, and development kits. The platform provides plug-in architecture for sharing components in a problem domain. DigitalEdge can be provisioned as a fully integrated solution on x86 hardware, or off-premise on a public cloud. DigitalEdge runs as a platform-as-a-service (PaaS) in:

- The Amazon AWS™ public cloud
- Amazon Virtual Private Cloud™ (VPC™)
- A private cloud using Eucalyptus® loaded across x86 servers in your own data center

Cloud computing provides on-demand network access to a shared pool of configurable resources such as servers, storage, applications, and services. Resources are automatically provisioned with minimal intervention, scaling up during peak times and scaling back as needs decrease.

The DigitalEdge Advantage

DigitalEdge can help you protect your enterprise's critical assets and enhance your business insight by ingesting and processing big data in real time and providing continuous analytics – all in one platform.



Real Time Analytics

DigitalEdge feeds a variety of consolidated data (which could be large, continuous streams, or static sets of data) into a configurable data model. Raw data is enriched with context and meaning from additional information sources. DigitalEdge can also process legacy data, to uncover patterns of an historical or ongoing significance. The system is optimized for queries and alerts, providing the analyst with near real-time data access and reports. In a matter of seconds - not minutes or hours - the raw data is parsed, processed, and available as actionable intelligence to the analyst, with little manual effort required of the analyst. DigitalEdge can scale to billions of records per day without any manual intervention. Latency between data parsing time to analytics is less than a minute.

DigitalEdge Features

DigitalEdge is a technologically advanced cloud platform. The system self-monitors its resources, automatically growing and shrinking in response to processing demands. DigitalEdge is also highly modular, providing common components and the tools to build custom components as needed. The DigitalEdge platform is optimized to deliver the best performance and the best results:

- **Real-time analytics**: Data is available in seconds, for querying and alerting to potential risks
- **Flexibility**: DigitalEdge runs on a public or a private cloud, and processes structured or unstructured data
- **Data enrichment**: Data is enhanced from diverse sources, adding context and meaning to raw data. There are different methods for adding context, such as correlating data with key lookups or via algorithms. But in all cases, the results are the same: all data is merged into one record. This results in faster queries and immediate context to facts for analysts.

- **Plug-in configurable architecture**: Plug-in components for parsers, enrichment processors, and data sinks can be shared, or configured for a private system
- **Dynamic scalability**: As incoming data increases exponentially, the system will scale horizontally without the need for human intervention; resources are elastic, increasing or decreasing to match processing loads
- **Automatic resource monitoring and provisioning**: Resources are allocated on-demand and usage is tracked automatically; clusters of servers and storage devices sized for a maximum capacity use case don't have to be maintained. Costs can be a lot less in the public cloud when you only pay for consumed resources.
- **Self-managed**: DigitalEdge Administrators use a set of tools to configure, run, and monitor their own systems. The multi-tenant system on a cloud platform runs a single instance of the software while serving many organizations or tenants.

DigitalEdge Use Cases

DigitalEdge can be applied to many different use cases involving security breaches, situational awareness, cybersecurity protection, fraud and insider threats, providing incident response and remediation. The software can be tailored to provide on-demand analytics of big data in a variety of industries:

- Financial firms: looking out for suspicious transactions, guarding against financial fraud, detecting possible money laundering, and protecting account credentials
- Telecommunications industry: analyzing firewall log files for potential security hacks, preventing distributed denial-of-service (DDoS) attacks, providing advanced persistent threat (APT) deterrence, and protecting against DNS abuse
- E-commerce vendors: safeguarding customer records, detecting security violations, and protecting against credit card fraud
- Transportation industry: taking advantage of the DigitalEdge geospatial analytics to track vehicles and packages, to analyze patterns in pathways, and to flag abnormal behavioral states
- Infrastructure organizations that rely upon sensor data: gaining intelligence from correlating sensor data, maintenance logs, and smart grids, identifying problematic situations and failing systems, determining the lifespan of critical parts

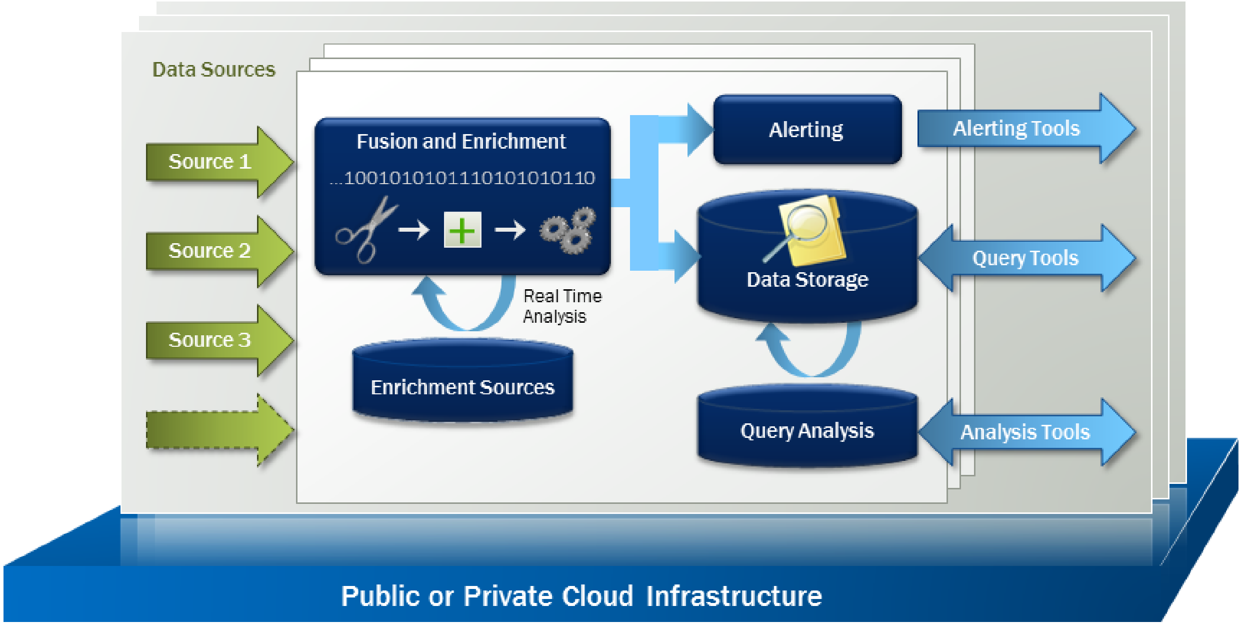
Product documentation

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
Overview Guide	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
Configuration Guide	Instructions for defining data models and building processing pipelines	Data Specialists, DigitalEdge Administrators
Operations Guide	Daily administration information, covering monitoring, managing, and modifying the platform	DigitalEdge Administrators
DigitalEdge SDK Guide	Reference for building custom plug-in components	Developers
Alerts API Guide	Reference for specifying data triggers and notifications for an alerting capability	Developers
Search API Guide	Reference for providing search services on a Lucene data sink node	Developers

System Overview

In a typical DigitalEdge system, raw data feeds enter via several different transports (protocols such as UDP, used for transferring large amounts of data over a network into DigitalEdge) from multiple data sources. Each data source can be in a different format. Using the DigitalEdge Setup Tools, you can configure input formats, a normalized data model, and parsers for site-specific needs. In the DigitalEdge processing pipeline, parsers extract the data, the fusion engine uses translations that you specify for normalizing the data into name-value pairs, and the enrichment engine adds context and meaning to the data with dimensional data and algorithmic enrichments. Processed data is then sent to data sinks to index the data and to post-process and store data for multiple uses. Data can also be sent to systems external to DigitalEdge. The alerting engine analyzes data and produces automatic alerts (console messages, text messages, emails, JMS topics) about potential threats. From the time that data is first parsed to the point when it is available for analysis, less than a minute has transpired.



Overview of the DigitalEdge System

Data Model

DigitalEdge is highly configurable to meet critical site-specific needs. Data configuration is the most important task in implementing an effective system. DigitalEdge provides Setup Tools to help you define the data model. A data model specifies how the data looks, how it is enriched, how input is mapped to it, and how dimensions are used to enrich the data.

Elements of the Data Model

The data model includes:

- **Input model**: A simple JSON-based model that specifies how the data will look when it enters the system and is mapped into a normalized data model. The input model can support multiple versions simultaneously. (JSON, or JavaScript Object Notation, is a text-based standard for data exchange that can represent data structures, arrays, and objects. JSON is easy to create, read, and parse.)
- **Data source mapping**: Specifications for parsing the incoming data and mapping each field to the input model.
- **Enrichments**: Rules for enhancing the data with additional context and meaning (such as replacing customer IDs with customer names, or translating latitude/longitude coordinates to zip codes).

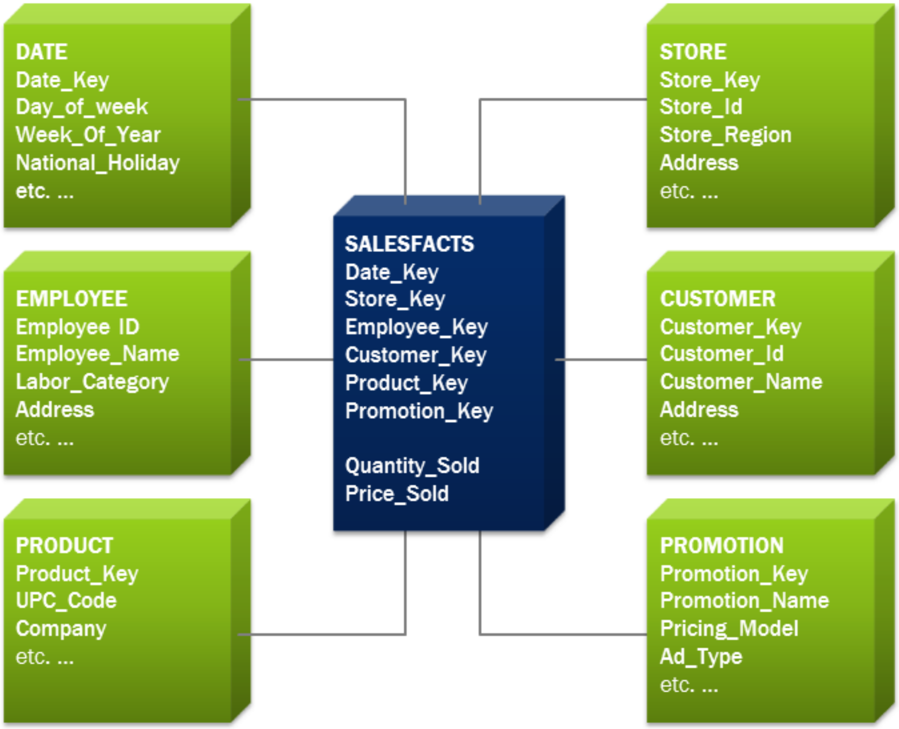
The DigitalEdge data model is optimized for query performance and analytics.

Dimensional Data Modeling and Data Enrichment

To achieve rapid response to situational information, data cannot be opaque, coded, missing, or resident in an inaccessible data store. DigitalEdge correlates data from multiple data sources up-front during ingest processing, resulting in better query performance and data records which integrate facts and key descriptive values all in one record. To achieve these benefits, DigitalEdge uses a dimensional data modeling methodology with dimension table enrichments.

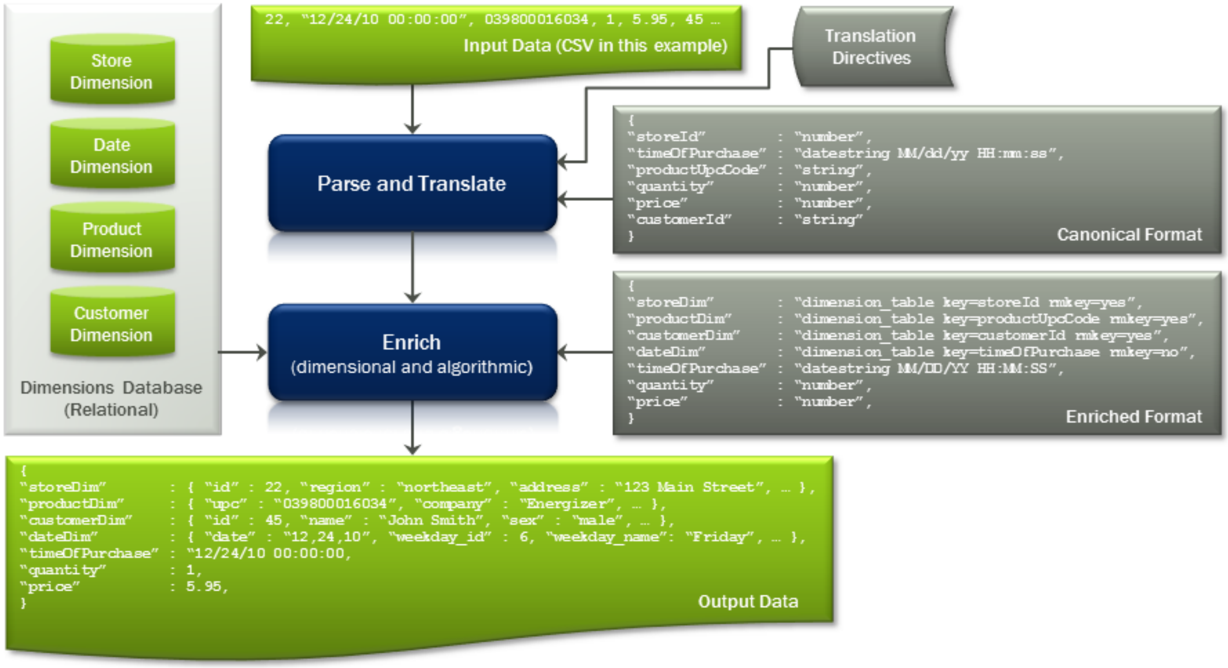
Dimensional data modeling (DDM) is a database concept that is different than the third normal form entity-relationship (ER) modeling in traditional relational databases. ER modeling, created for transactional systems, was often complex, hard for users to understand, difficult for analytic tools to use, and required large joins that often resulted in very poor performance. DDM was created to support data querying performance and data understandability. DDM uses a standard structure that is readily understood by users and analytic tools, results in faster queries, provides descriptions for facts, and can be relatively easy to change. DDM excels with real time data processing and large scale analytics.

In DDM, a *fact* is usually a measurement or numeric value such as a timestamp, a sales price, or an account ID; a *dimension* is a category of data that provides descriptive and contextual information for the main data feed/facts. For example, in a point of sale system, a sales record (the fact) is *enriched* with contextual customer data, store information, employee details, and product data (the dimensions). Dimensions improve the usability and enhance the meaning of raw data by adding additional information to the standard input model.



Dimensional Data Model

Traditional RDBMS implement DDM with foreign keys in the main table, referencing rows in the dimension tables. To integrate the data, database joins are performed during query time so the customer name (not the customer ID) is available in the retrieved record. But doing joins during a user's search can significantly reduce performance. In a NoSQL environment, joins are not possible. So DigitalEdge pre-joins the dimension records to the raw data at ingest time. In other words, all the data from the customer record is merged with the sales record, eliminating the need for joining records or accessing other databases when querying records. All relevant data is pre-joined and stored in one record; the original record, which may have had just a few numbers, ids, or codes in it, is now enhanced with inline data. This process of correlating fact records with dimensional records is called *dimension table enrichment*, *keyed dimensional enrichment processing*, or simply *enrichment*. By pre-joining facts and dimensional data and storing all relevant data in one record, enrichment contributes to faster real-time queries and analytics.



```
22, “12/24/10 00:00:00”, 039800016034, 1, 5.95, 45 …
```

```
{
“storeId”         : “number”,
“timeOfPurchase”  : “datestring MM/dd/yy HH:mm:ss”,
“productUpcCode”  : “string”,
“quantity”        : “number”,
“price”           : “number”,
“customerId”      : “string”
}
```

```
{
“storeDim”        : “dimension_table key=storeId rmkey=yes”,
“productDim”      : “dimension_table key=productUpcCode rmkey=yes”,
“customerDim”     : “dimension_table key=customerId rmkey=yes”,
“dateDim”         : “dimension_table key=timeOfPurchase rmkey=no”,
“timeOfPurchase”  : “datestring MM/DD/YY HH:MM:SS”,
“quantity”        : “number”,
“price”           : “number”,
}
```

```
{
“storeDim”        : { “id” : 22, “region” : “northeast”, “address” : “123 Main Street”, … },
“productDim”      : { “upc” : “039800016034”, “company” : “Energizer”, … },
“customerDim”     : { “id” : 45, “name” : “John Smith”, “sex” : “male”, … },
“dateDim”         : { “date” : “12,24,10”, “weekday_id” : 6, “weekday_name”: “Friday”, … },
“timeOfPurchase”  : “12/24/10 00:00:00,
“quantity”        : 1,
“price”           : 5.95,
}
```

Dimension Table Enrichment

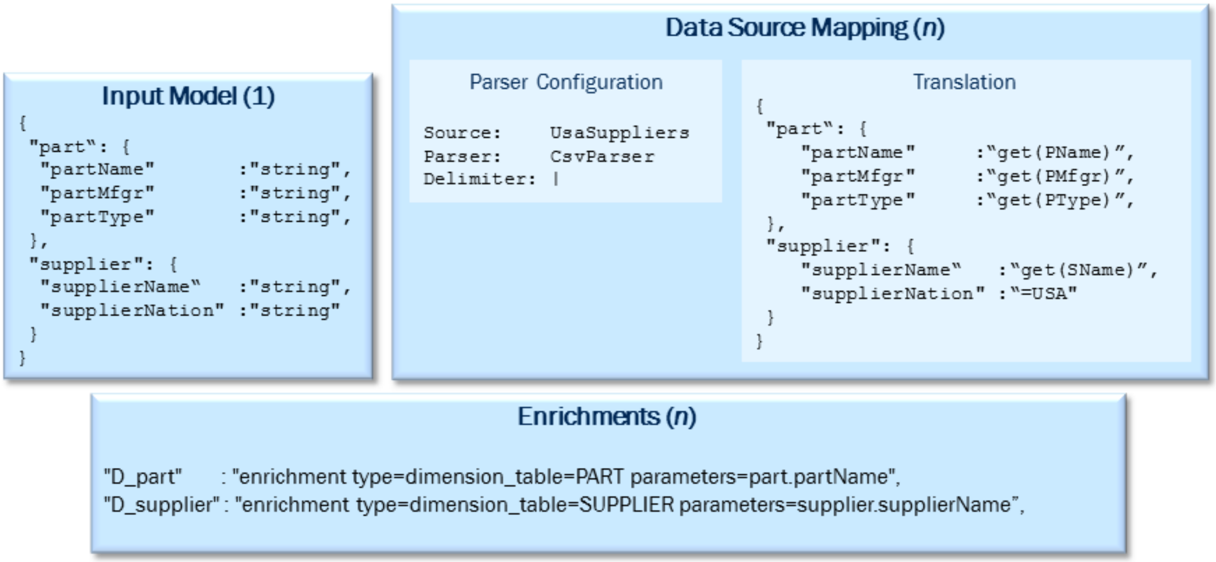
Besides dimension table enrichment, DigitalEdge provides other algorithmic enrichment techniques, known as *generalized enrichments*. Generalized enrichment extends DDM pre-joins to standard data available from other data sources. For example, one algorithm computes the nearest zip code given a set of latitude and longitude coordinates; another determines if a given IP address is local or private.

Data enrichment components can be private, applicable only to your system, or they can be shared with peers in a vertical market. For example, a set of data components could be developed for cyber security threats and shared among banks for a coordinated defense.

See "Enrichment processors" on page 16 for a complete list of enrichment components provided with DigitalEdge.

Data Model Example

Here is an illustration of the three parts of the data model: the input model, the data source mappings, and the enrichments.



Input Model (1)

```
{
 "part": {
  "partName"         :"string",
  "partMfgr"         :"string",
  "partType"         :"string",
 },
 "supplier": {
  "supplierName"     :"string",
  "supplierNation"   :"string"
 }
}
```

Data Source Mapping (*n*)

Parser Configuration

```
Source:     UsaSuppliers
Parser:     CsvParser
Delimiter:  |
```

Translation

```
{
 "part": {
    "partName"       :"get(PName)",
    "partMfgr"       :"get(PMfgr)",
    "partType"       :"get(PType)",
 },
 "supplier": {
    "supplierName"    :"get(SName)",
    "supplierNation"  :"=USA"
 }
}
```

Enrichments (*n*)

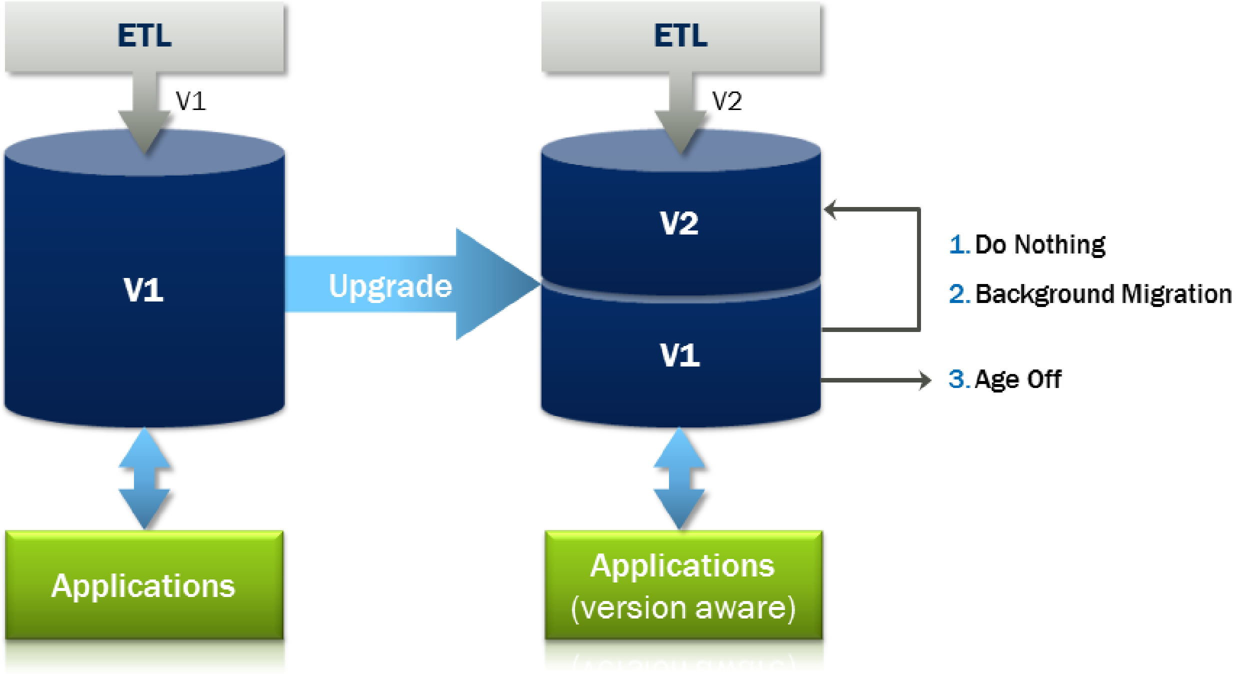
```
"D_part"     : "enrichment type=dimension_table=PART parameters=part.partName",
"D_supplier" : "enrichment type=dimension_table=SUPPLIER parameters=supplier.supplierName",
```

Three Parts of the Data Model

In this example, the CSV parser extracts pipe delimited data. It is mapped to the input model (on the left) using the translation rules (on the right). For example, the constant, “USA”, is assigned to the “supplierNation” field. At the bottom of the figure, two enrichments are specified: one to look up the part name in the part database and to insert the name into the D_part field; the other to add the supplier’s name into the D_supplier field.

Data Model Versioning

A data model represents an understanding of the data (data semantics) and its use (data syntax). In traditional RDBMS, a data model is fixed and static. Changes often require complex migrations and significant modifications to code and applications. In a NoSQL environment, because there is no schema defining the data model, multiple versions of the data model can exist simultaneously.



Data Model Versioning

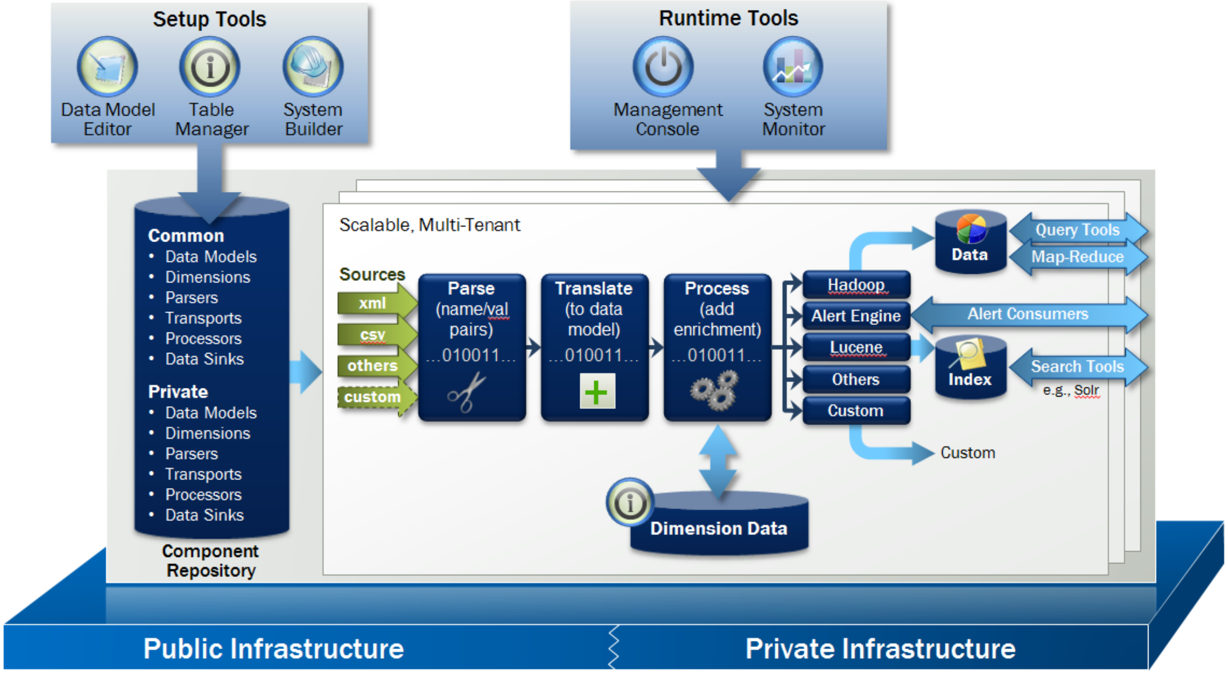
DigitalEdge uses data model versioning to accommodate multiple data models. Every record in the system is tagged with its data model and version, allowing multiple versions to exist in the same data store until you are ready to migrate version 1 data to version 2, or until version 1 data is ready to age off after a specified number of days. Versioning also makes data migrations much quicker, with less operational impact.

Architecture

DigitalEdge is highly modular and customizable. Its plug-in architecture uses shared components that can be swapped in and out, configured, and used in several different deployments. Plug-ins are stored in the System Repository.

It is critical to devote enough time to configuration activities to implement a superior system. Configuration specifies the input model, parsing rules, data source mapping into the input model, enrichment and processing needs, data sink specifications, and alerting rules. DigitalEdge configuration may require the services of several individuals, such as a Data Specialist, a Database Administrator, a DigitalEdge Administrator, a Business Analyst, or a Developer.

DigitalEdge provides several interface tools to configure and manage your system. Setup Tools are the brains of the system, specifying the details of the System Repository, plug-in configuration, and the processing pipeline. Data specialists use these tools heavily during the configuration phase. Once a system is up and running, the Runtime Tool set is the heart of the system, for monitoring resource consumption, automatic scaling of all resources, and keeping data flowing.



DigitalEdge Architecture

DigitalEdge can also provide user tools to facilitate real time analytics. User tools are the eyes and ears of the system, bringing potential problems, anomalies, and actionable issues to your attention.

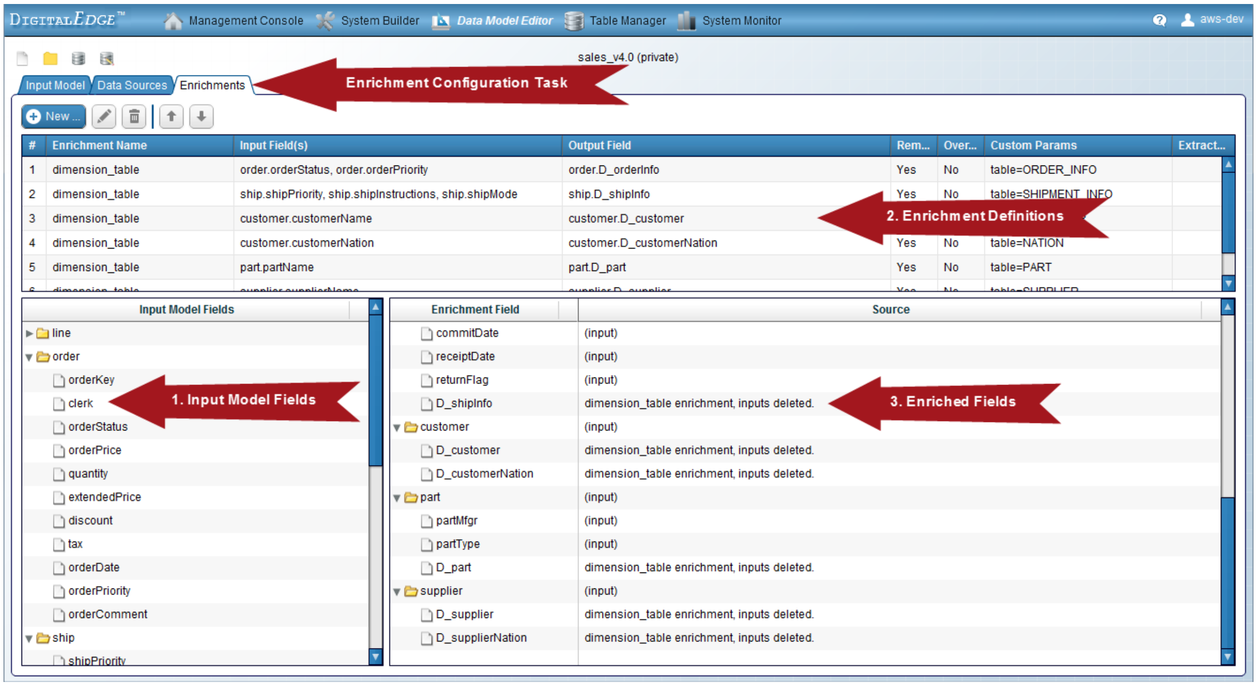
Configuration Tools (Setup Tools)

The DigitalEdge Setup Tools help you specify data models, dimension data, and data processing rules. These tools are used during initial system configuration and customization; once the system is up and running, the tools are only needed for occasional modifications and refinements.

Data Model Editor

The Data Model Editor is used heavily during data and system configuration. Use it to configure a complete data model, or use its **Data Model Creation Wizard** to create a simple data model (without nesting or enrichments) from a flat CSV or JSON file.

- **Users**:
  - Data Specialists, Database Administrators
- **Uses**: Data configuration tasks, including:
  - Define the input model: Specify fields that all data sources will be standardized to
  - Specify data source formats for the Parse stage: Identify data source tables and fields to parse and extract
  - Map raw source data to the input model for the Translate stage: Create translation statements, mapping extracted fields to the input model
  - Define enrichments for the Process stage: Specify dimension tables and fields, then create enrichment definitions to fuse dimension data with the input model
- **Results**: A master data model for the system
- **Example**: This screen depicts the results of the enrichment configuration task.
  1. Input model fields were selected for enrichment
  2. Specifications were input for each enrichment
  3. Resulting enriched fields are displayed for final review



sales_v4.0 (private)

Enrichment Configuration Task

#	Enrichment Name	Input Field(s)	Output Field	Rem...	Over...	Custom Params	Extract...
1	dimension_table	order.orderStatus, order.orderPriority	order.D_orderInfo	Yes	No	table=ORDER_INFO	
2	dimension_table	ship.shipPriority, ship.shipInstructions, ship.shipMode	ship.D_shipInfo	Yes	No	table=SHIPMENT_INFO	
3	dimension_table	customer.customerName	customer.D_customer				
4	dimension_table	customer.customerNation	customer.D_customerNation	Yes	No	table=NATION	
5	dimension_table	part.partName	part.D_part	Yes	No	table=PART	
6	dimension_table	supplier.supplierName	supplier.D_supplier	Yes	No	table=SUPPLIER	

2. Enrichment Definitions

Input Model Fields
▶ line
▼ order
orderKey
clerk
orderStatus
orderPrice
quantity
extendedPrice
discount
tax
orderDate
orderPriority
orderComment
▼ ship
shipPriority

1. Input Model Fields

Enrichment Field	Source
commitDate	(input)
receiptDate	(input)
returnFlag	(input)
D_shipInfo	dimension_table enrichment, inputs deleted.
▼ customer	(input)
D_customer	dimension_table enrichment, inputs deleted.
D_customerNation	dimension_table enrichment, inputs deleted.
▼ part	(input)
partMfgr	(input)
partType	(input)
D_part	dimension_table enrichment, inputs deleted.
▼ supplier	(input)
D_supplier	dimension_table enrichment, inputs deleted.
D_supplierNation	dimension_table enrichment, inputs deleted.

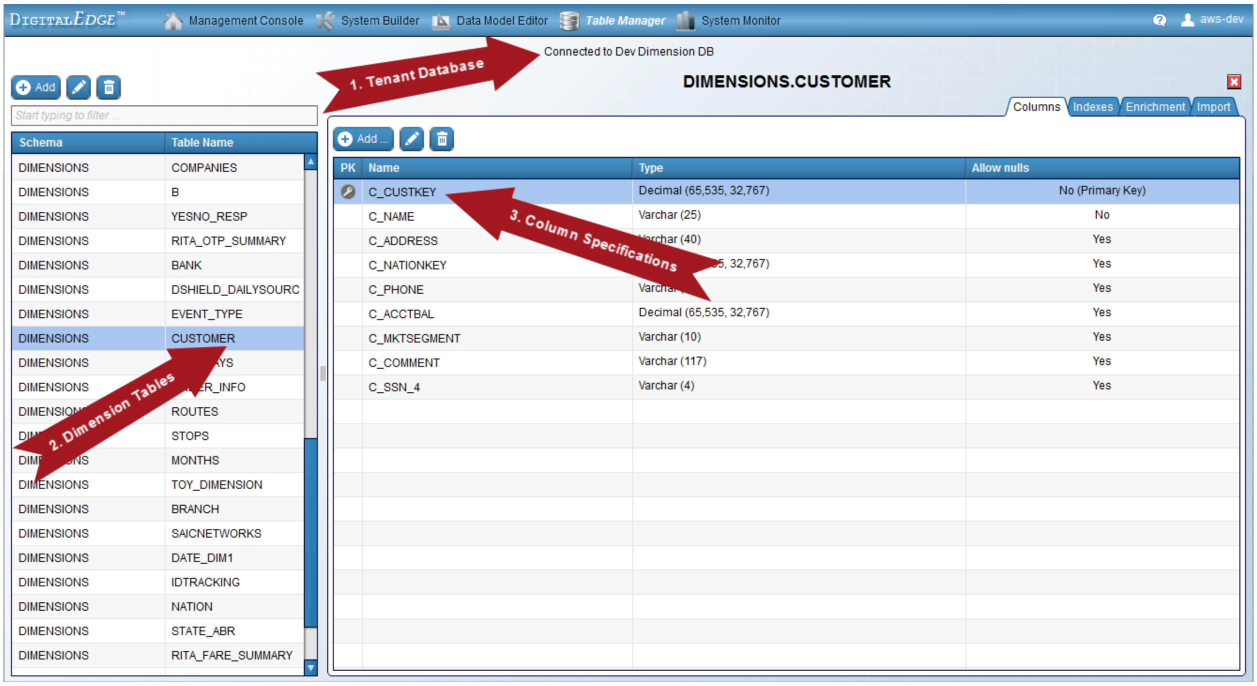
3. Enriched Fields

Table Manager

The Table Manager is currently used to define and edit enrichment sources (dimension tables) for the Process stage.

- **Users**:
  - Data Specialists
  - Database Administrators

- **Uses**: Define data tables and columns that will be used in enrichments
- **Results**: Fully specified dimension tables and columns
- **Example**: This screen depicts the results of fully defining a database table and its fields that may be used for enrichment:
  1. Tenant database was created
  2. Dimension tables were added
  3. Columns were specified for all the dimension tables



1. Tenant Database

2. Dimension Tables

3. Column Specifications

Connected to Dev Dimension DB

DIMENSIONS.CUSTOMER

Schema	Table Name
DIMENSIONS	COMPANIES
DIMENSIONS	B
DIMENSIONS	YESNO_RESP
DIMENSIONS	RITA_OTP_SUMMARY
DIMENSIONS	BANK
DIMENSIONS	DSHIELD_DAILYSOURC
DIMENSIONS	EVENT_TYPE
DIMENSIONS	CUSTOMER
DIMENSIONS	YS
DIMENSIONS	ER_INFO
DIMENSIO	ROUTES
DI	STOPS
DIM NS	MONTHS
DIMENSIONS	TOY_DIMENSION
DIMENSIONS	BRANCH
DIMENSIONS	SAICNETWORKS
DIMENSIONS	DATE_DIM1
DIMENSIONS	IDTRACKING
DIMENSIONS	NATION
DIMENSIONS	STATE_ABR
DIMENSIONS	RITA_FARE_SUMMARY

PK	Name	Type	Allow nulls
🔑	C_CUSTKEY	Decimal (65,535, 32,767)	No (Primary Key)
	C_NAME	Varchar (25)	No
	C_ADDRESS	archar (40)	Yes
	C_NATIONKEY	5, 32,767)	Yes
	C_PHONE	Varchar	Yes
	C_ACCTBAL	Decimal (65,535, 32,767)	Yes
	C_MKTSEGMENT	Varchar (10)	Yes
	C_COMMENT	Varchar (117)	Yes
	C_SSN_4	Varchar (4)	Yes

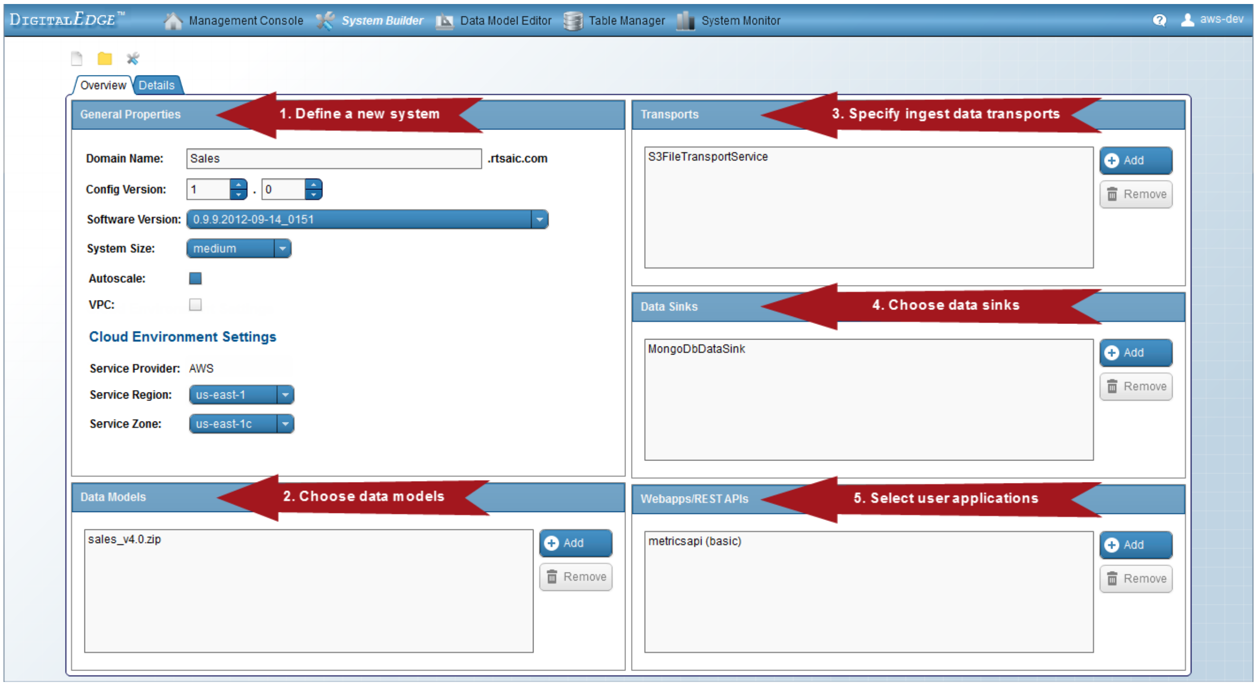
System Builder

The System Builder is used to define a new system and assemble its components.

- **Users**: DigitalEdge Administrators
- **Uses**:
  - Define a system

  - Specify the plug-in components (data model, transports, data sinks, etc.)
  - Assemble the processing pipeline

- **Results**: A complete system configuration that is ready to run
- **Example**: This screen depicts a newly specified system and processing pipeline:
  1. System definition and sizing
  2. Data model
  3. Ingest transports
  4. Data sinks
  5. User applications



General Properties — 1. Define a new system

Domain Name: Sales .rtsaic.com

Config Version: 1 . 0

Software Version: 0.9.9.2012-09-14_0151

System Size: medium

Autoscale: ☑

VPC: □

Cloud Environment Settings

Service Provider: AWS

Service Region: us-east-1

Service Zone: us-east-1c

Data Models — 2. Choose data models

sales_v4.0.zip

Transports — 3. Specify ingest data transports

S3FileTransportService

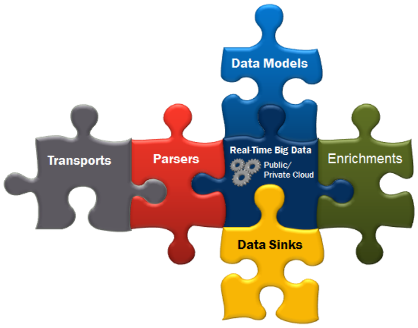
Data Sinks — 4. Choose data sinks

MongoDbDataSink

Webapps/REST APIs — 5. Select user applications

metricsapi (basic)

Plug-in Architecture



The DigitalEdge platform is flexible and modular, accommodating private or shared plug-ins. For example, plug-ins developed for a specific industry (e.g., financial) can be shared and reused across different deployments. Plug-ins can be integrated at run-time. Plug-ins can be created for:

- **Transports**: protocols used for transferring large amounts of data over a network into the DigitalEdge system
- **Parsers**: to translate source data into key-value pairs

- **Input models**: specifying how the data will look in DigitalEdge after normalization and data source mapping
- **Enrichments**: to perform data enrichment processing on a data stream
- **Data sinks**: to store, index, or post-process pipeline data

DigitalEdge includes a repository of shared and private plug-ins. DigitalEdge supplies a library of plug-ins that can be shared in a common ecosystem. Common plug-ins can be modified and saved as customized components for a private system. DigitalEdge stores all private and configured components in the Master Repository, pulling them from the Master Repository when a new system is implemented.

Transports

The transports define mechanisms for getting DigitalEdge data from data sources. A transport simply wraps the data in a JMS message, it does not convert data. Pre-defined transports include (but are not limited to):

- **Database Watcher**: Gets data from a database by running an SQL select query
- **Directory Watcher**: Polls a local or remote file system directory for files
- **Directory Crawler**: A beta transport that crawls and processes data in a local or remote file system while decompressing zipped files and processing files that match wild card patterns
- **Hive**: Transports content from a Hive data sink to another data sink
- **JMS Bridge**: Copies JMS messages between two JMS servers (Java Message System is a Java API for sending messages between clients)
- **MongoDB**: Transports content from a MongoDB data sink to another data sink
- **PCap**: Transports pcap packets on a specified network interface
- **S3**: Transports files sent to an Amazon S3™ bucket
- **TCP**: Transports the contents of an entire TCP stream (Transmission Control Protocol/Internet Protocol is a core protocol for the Internet, sending a stream of data bytes from one computer to another)
- **Twitter transports**: Several transports use the Twitter APIs to search the Twitter stream for specific Tweets
- **UDP**: Transports UDP packets to a configured port (the User Datagram Protocol is an alternative to TCP which retrieves data units traveling between computers, but which does not divide the message into packets at the receiving end)
- **URL**: Reads the contents of a URL (such as an RSS feed) and puts the data on the JMS input queue

You can also use the **Data Transfer Utility** as a simplified path for getting data into DigitalEdge from from your desktop or from a mapped network drive.

Parsers

Parsers extract and translate data sources into the input model's name-value pairs. Parsers work with many different formats, and include (but are not limited to):

- **Binary Parser**: A configurable parser that works with binary files that must be parsed and mapped to readable data for the application. The parser is configurable to extract specific data fields from your input source.
- **CEF**: This parser works with the Common Event Format from ArcSight®, an open standard for logging security-related information across various devices and applications in a common event log format
- **CSV**: A configurable parser that extracts delimited plain text; delimiters include comma, pipe, and new line
- **DNS**: A parser that reads and extracts DNS packets
- **Email**: This parser works with the RFC 822 format, a standard format from ARPA that specifies email text message structures, including the to and from fields, attachments, etc.
- **EXIF**: A configurable parser that works off the Exchangeable Image File Format, used for handling image and sound files from digital cameras
- **JSON**: A configurable parser that works with JavaScript Object Notation, a text-based standard for data exchange, to extract fields, arrays, and objects
- **Libpcap**: This parser works with packet captures in the UNIX Libpcap library
- **Log File**: A parser that reads any log file, extracts the timestamp, and saves the remaining string as the substantive log entry
- **SNMP**: This parser reads and extracts Simple Network Management Protocol packets (SNMPv1 or SNMPv2)
- **Unstructured file**: a beta parser that works with unstructured files such as Word, Excel, and PDFs to extract content and metadata
- **XML**: A configurable parser for the Extensible Markup Language, often used to represent arbitrary data structures

Enrichment processors

The enrichment processors add context and meaning to the incoming data by enhancing the raw data with dimension data. You can define your own dimensional enrichments, using tables from specific data sources to enhance raw data, or you can apply generalized, algorithmic enrichments that are available as common components. Enrichments include (but are not limited to):

- **Dimension table enrichment**: Table lookups use specified natural key fields for exact matches on dimension records
- **Fuzzy match**: A beta processor that enriches records with a standardized string based on a fuzzy match lookup on a specified input field
- **Generalized enrichments**:

- **IP network**: Location data is determined from an IP address
- **Math enrichment**: Runs a mathematical expression that you define against the data
- **Postal location**: Latitude/longitude data is used to determine the nearest postal code, city, state, or country
- **Record history**: Adds an array of records that passed through DigitalEdge and that match a field
- **Regex**: Extracts sub-fields from a more complex input feld
- **SQL select**: Looks up information in an SQL database with a given query

Data sinks

A data sink is queue, server, or database that can receive pipeline-processed data to store or post-process for other uses. DigitalEdge includes NoSQL ("Not Only SQL") data stores which are built specifically to handle big data and scaling. Characteristics of NoSQL include:

- Horizontally scalable on commodity servers or in the cloud, to handle big data at a reasonable cost
- Inherently distributed
- Featuring MapReduce functionality to parallelize queries and analytical processing. In MapReduce systems, each server processes a portion of a problem ("map"), and then the results are combined into the final result ("reduce").
- Schema-free or schema flexible to support various input formats
- Absent of complex and slow joins

Processed data is sent to one of several types of data sinks:

- Alerting data sink: a filtering engine applies rules to the data and passes alert criteria to users when specific conditions are met which raise red flags in the data
- Back-end data store: a NoSQL database that stores data for other uses, for the historical record, or for indexing

Plug-in data sinks include (but are not limited to):

- **Alerting Data Sink**: The DigitalEdge alerting engine applies filters to incoming data and sends an alert to a JMS topic or an email to a user when a match is found. You can also configure an automatic response to an alert.
- **Cassandra Data Sink**: A beta data sink that stores data in an Apache Cassandra cluster
- **Dimension Data Sink**: Stores dimension table data for enrichments in a customer's tenant database
- **Elasticsearch Data Sink**: A beta data sink that stores and indexes data for full text search with Elasticsearch

- **External HBase Data Sink**: Stores data in an HBase cluster located outside of DigitalEdge
- **External HDFS Data Sink**: Stores data in a Hadoop cluster located outside of DigitalEdge
- **External Hive Data Sink**: Stores data in the Hadoop/Hive environment outside of DigitalEdge
- **HBase**: Stores data in an HBase database
- **Hive**: Stores data in the Hadoop/Hive environment, managed by DigitalEdge
- **JSON to JDBC Data Sink**: Maps JSON objects to a relational database table and writes them to a database with JDBC
- **Lucene Indexing Data Sink**: Indexes data for real-time and near real-time search
- **MongoDB**: Stores the resulting data in a MongoDB® database. MongoDB is a JSON document store, providing native JSON record storage and extensive query language support.
- **Sleep Data Sink**: Continually reads a record and then sleeps for a specified amount of time

Master Node

When configuration is complete, you create a new DigitalEdge system by running it from the Management Console, which starts up the Master Node. The Master Node has the ability to start and stop all instances in the system, to monitor use of all resources, and to make automatic scaling decisions. The Master Node:

- Handles virtual storage allocations
- Controls automatic scaling
- Starts up and shuts down instances in an orderly manner
- Gathers performance metrics as input into scaling decisions
- Adds or removes nodes based on load and storage utilization

The Master Node uses several parameters created during configuration to control dynamic scaling, including the initial startup allocations, a scaling condition for all resources, and a scale down condition to conserve resources and costs.

Dynamic Scaling

Whether in a public or private cloud, DigitalEdge runs a virtualized environment with resources automatically provisioned as needed. Rather than configuring for peak loads, the Master Node accommodates spikes and drop-offs in activity by scaling resources dynamically to meet any load demand. For example, you can sustain the throughput rate required to maintain the data flow rate set by your input sources. The Master Node automatically starts virtual machines, allocates virtual storage, and establishes network parameters for optimum system operation. Changes in both storage and processing loads are handled gracefully without human intervention or the installation of additional hardware.

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. Coupled with CAS, DigitalEdge is built upon Amazon Web Services™ (AWS™)/Eucalyptus®, leveraging a common set of APIs for management tools across the platforms. 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. System backup and recovery services are provided at a customized level.

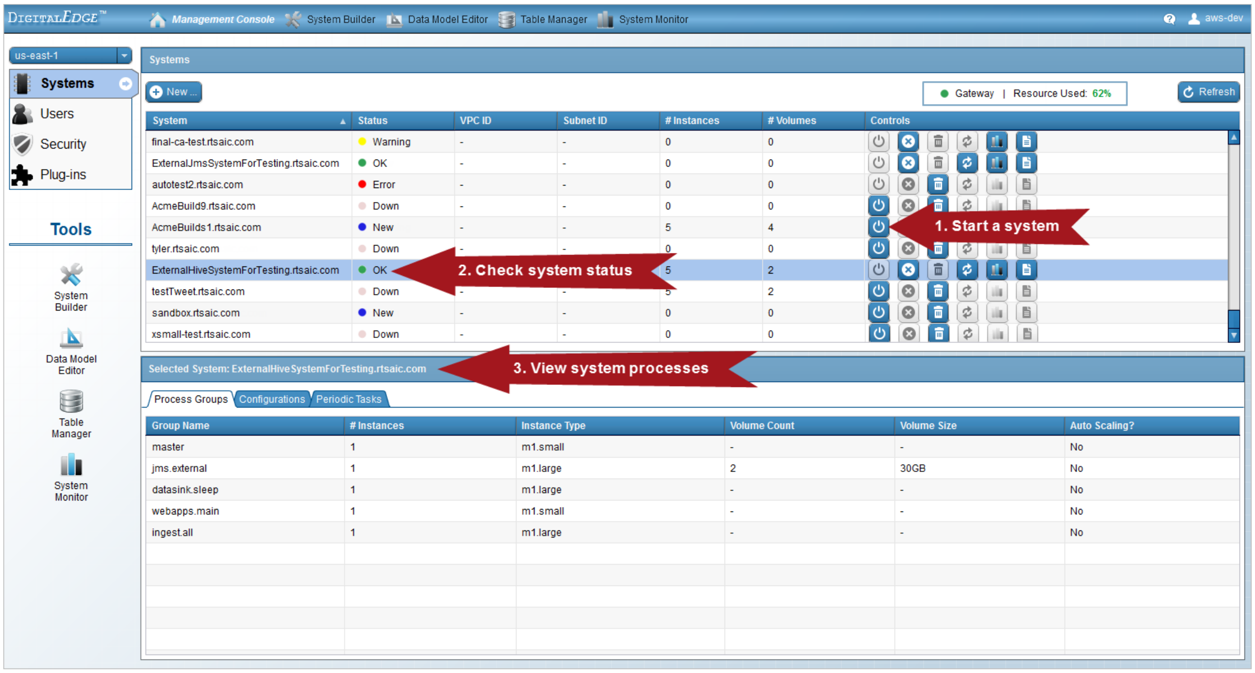
Management Tools (Runtime Tools)

After you have configured the DigitalEdge system, use the Runtime Tools on a daily basis to manage the system and to monitor resource use.

Management Console

The Management Console is used to start/stop a system and to manage system security. It is also a central location for accessing all the setup and runtime tools in DigitalEdge.

- **Users**: DigitalEdge Administrators
- **Uses**: System management tasks:
    - Access all the UI tools
    - Start and stop systems
    - View a snapshot of system status
    - Create and manage users
    - Manage security groups and rules
- **Results**: A running system
- **Example**: This screen depicts a running system and a snapshot of its current status:
    1. System was started
    2. System is running
    3. Process groups are specified



Systems

● Gateway | Resource Used: 62%

System	Status	VPC ID	Subnet ID	# Instances	# Volumes	Controls
final-ca-test.rtsaic.com	Warning	-	-	0	0	
ExternalJmsSystemForTesting.rtsaic.com	OK	-	-	0	0	
autotest2.rtsaic.com	Error	-	-	0	0	
AcmeBuild9.rtsaic.com	Down	-	-	0	0	
AcmeBuilds1.rtsaic.com	New	-	-	5	4	
tyler.rtsaic.com	Down	-	-	0	0	
ExternalHiveSystemForTesting.rtsaic.com	OK			5	2	
testTweet.rtsaic.com	Down	-	-	5	2	
sandbox.rtsaic.com	New	-	-	0	0	
xsmall-test.rtsaic.com	Down	-	-	0	0	

Selected System: ExternalHiveSystemForTesting.rtsaic.com

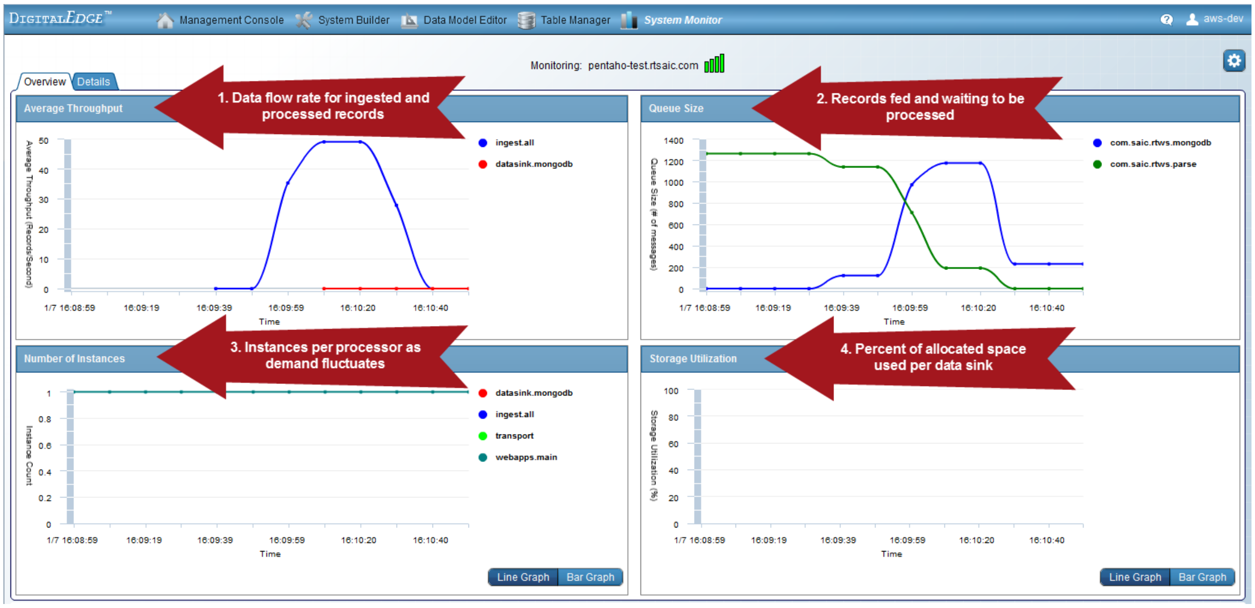
Process Groups | Configurations | Periodic Tasks

Group Name	# Instances	Instance Type	Volume Count	Volume Size	Auto Scaling?
master	1	m1.small	-	-	No
jms.external	1	m1.large	2	30GB	No
datasink.sleep	1	m1.large	-	-	No
webapps.main	1	m1.small	-	-	No
ingest.all	1	m1.large	-	-	No

System Monitor

Once a system is up and running, the System Monitor visually depicts system activity and resource scaling. You can watch processes auto-scale up and down as resource utilization changes.

- **Users**: DigitalEdge Administrators
- **Uses**: Monitor system operations visually:
    - Check data flow rates
        - Ingest rate
        - Record processing backlog
        - Alerting engine throughput
    - Monitor resource consumption and auto-scaling for:
        - Data storage
        - Record processing
        - User applications
- **Results**: Dynamic picture of system health, resource consumption, and problems
- **Example**: The Overview graphs depict a running system and four monitored areas:
    1. Average Throughput: The rate at which ingested records are fed into the system
    2. Queue Size: The number of records that were fed into the system but which are not yet been processed
    3. Number of Instances: The number of instances that are automatically allocated and consumed for each process
    4. Storage Utilization: Percent of allocated space currently used by EBS (Amazon Elastic Block Storage) data sinks



1. Data flow rate for ingested and processed records

2. Records fed and waiting to be processed

3. Instances per processor as demand fluctuates

4. Percent of allocated space used per data sink

User Tools

Data is ingested, processed, and available for searching and analysis in a matter of seconds.

DigitalEdge includes several tools for presenting data to end users. You can also build your own user tools, export DigitalEdge processed data to non-DigitalEdge systems, run external reporting programs on DigitalEdge data, etc. DigitalEdge user tools include (but are not limited to):

- **Searching**: Users can search DigitalEdge processed data in real time with full text query capabilities
- **Alerting**: Business rules in the alerting engine can notify an administrator of an anomalous event as it occurs, enabling proactive measures to be taken in a timely manner. Alerts can be generated as email messages, text messages, on a mobile device, or in a JMS queue.
- **Pentaho server**: Processed data can be presented in graphical dashboards, self-service reporting, and visual representations with this readily-available open source product

DigitalEdge can also perform historical analysis of legacy data. This can be particularly useful to data scientists who are developing analytical algorithms and refining alert parameters. Historical data analysis is also useful in supporting root cause analysis of incidents.

Appendix A: Terminology

Term	Definition
Amazon 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
Amazon S3™ (Simple Storage Service™)	The online storage web service provided with AWS™ and used as a data source for public cloud instantiations
AWS™ (Amazon Web Services™)	Remote web services that comprise the cloud computing platform offered by Amazon.com; AWS™ includes Amazon EC2™ and Amazon S3™
Big data	Extremely large data streams that are too big and awkward to process and analyze with traditional database management tools
Blade server	A streamlined server computer in a blade center that is optimized to save space and energy over traditional rack mounted servers
CEF (Common Event Format)	An open standard for logging security-related information across various devices and applications in a common event log format for interoperability, created by ArcSight, an HP company
Cloud computing	Delivery of software services and shared resources over the Internet as measured services; characterized by resource pooling, dynamic scaling of resources, elasticity, self-service, and on-demand access
Data model	The DigitalEdge data model includes: • The input model: a JSON specification of a normalized data model • The data source mappings: specifications for parsing the incoming data and mapping each field to the input model • Enrichment processing: rules for enhancing the data with additional context and meaning
Data Model Editor	A setup interface tool used to configure:

Term	Definition
	• Input Model • Data source mappings • Enrichments
Data sink	A queue, server, or database that can receive pipeline-processed JSON data to store or post-process for other uses
Dimension table enrichment	Also known as keyed dimensional enrichment; the technique of using natural dimension keys for table lookups to find exact matches on dimension records for data enrichment
Dimensional Data Modeling (DDM)	A technique in data warehouse design that differs from entity-relationship (ER) modeling in traditional relational databases. DDM excels with real time data processing and large scale analytics. DDM enhances data querying performance and data understandability, results in faster queries, provides context to facts, and offers the ability to change data models (schemas) easily. In DDM, a fact is usually a measurement or numeric value; a dimension is a category of data that provides descriptive meaning and contextual.
Enrichment	The process of merging all related information into one record, providing context for source data and enabling rapid analysis
ETL (Extract, transform, and load)	A process in data warehousing to collect data from external sources, to process the data for internal needs, and to load the data into a target system
Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems)	An open source software platform to implement private cloud computing. Eucalyptus implements the Amazon Web Services™ API (AWS) and provides its own command line tools, Euca2ools. Eucalyptus is compatible with most Linux® distributions, and can host Windows images.
Input model	A simple data model that specifies how the data will look when it enters DigitalEdge, is normalized, and is mapped into a data model
JSON (JavaScript Object Notation)	A text-based open standard for data exchange, derived from JavaScript, representing data structures, arrays, and objects. The DigitalEdge Data

Term	Definition
	Model is specified in JSON.
Management Console	A runtime interface tool used to start up the Master Node, to manage users and security groups, to view a snapshot of the system's status, to start and shut down systems
MapReduce	A term coined by Google, Inc. describing a framework for processing distributed problems across large datasets using many computer nodes and clusters (multiple nodes). The Map phase portions out the input data into small sub-problems and distributes them to worker nodes, in a tree structure. The Reduce phase is done by the master node, which combines answers from the sub-problems into a final result. As an example, for the query "what checks were deposited since noon today?", the mapping phase would have multiple jobs checking the transaction log on each of the active server nodes, and the reduce phase would compile and merge a master list of checks from all the sub-jobs for a final result.
Master node	The first node that runs in DigitalEdge, starting up, scaling, and shutting down all other nodes
NoSQL (Not Only SQL)	A class of DBMS that differ from traditional relational DBMS primarily because they do not use SQL as a query language. They do not use fixed schemas, joins, or vertical scaling.
Parser	A tool to extract fields from an input data source and translate them to key-value pairs; based on a file format type (binary, CSV, email, JSON, log files, XML, etc.)
Repository	The storage location for all plug-in components; the Master Repository is at the TMS level, the System Repository is at a tenant DigitalEdge account level
RFC 822	A standard format from ARPA that specifies email message structures, such as the to and from fields, and attachments
SIEM (Security Information and Event Management)	Technology that provides real time analysis of security data and alerts generated by networked applications; SIEMS are typically implemented in relational databases on commodity servers

Term	Definition
Splitter	Each transport works with a specific incoming record type (JSON, XML, PCAP, etc.); the tranport's record-format parameter uses a splitter to define record boundaries when the input data includes multiple records
System Builder	A setup interface tool used to specify the plug-ins (parsers, translators, processors, data sinks) in the DigitalEdge pipeline and to assemble the system
System Monitor	A runtime interface tool used to visually monitor data flow and resource consumption
Table Manager	A setup interface tool used to manage enrichment processing sources (dimension data) and other application-level tables
Tenant Management System (TMS)	Behind-the-scenes system that manages a DigitalEdge tenant's Setup tools, Runtime tools, repository, Master Node, APIs, databases, and security
Transport	A mechanism used for transferring large amounts of data over a network into the DigitalEdge system

Index