



# Product Overview






**Technical Summary, Samples, and Specifications**



# Table of Contents

<b>Introduction</b>	<b>4</b>
<b>Sample Uses</b>	<b>4</b>
<b>Voracity Operations</b>	<b>5</b>
<b>Data Discovery</b> 	<b>7</b>
<b>Data Classification</b>	<b>8</b>
<b>ER Diagrams</b>	<b>9</b>
<b>Database Profiling</b>	<b>9</b>
<b>Flat-File Profiling</b>	<b>10</b>
<b>Dark Data Discovery</b>	<b>10</b>
<b>Structured Metadata Discovery</b>	<b>10</b>
<b>Data Integration</b> 	<b>11</b>
<b>Single-Pass ETL</b>	<b>12</b>
<b>Legacy ETL Tool Acceleration &amp; Augmentation</b>	<b>13</b>
<b>Legacy ETL Tool Migration</b>	<b>13</b>
<b>Change Data Capture</b>	<b>14</b>
<b>Slowly Changing Dimensions (SCD)</b>	<b>14</b>
<b>Data Federation</b>	<b>15</b>
<b>Public/Private Mashups</b>	<b>15</b>

<b>Data Migration</b> 	<b>16</b>
Data-Type, File-Format & Database Migration	17
Endian Migration	17
Schema Migration	18
Data Replication	18
<b>Data Governance</b> 	<b>19</b>
Business Goals, Policy, and Rules	20
Data Quality	21
Data Masking	22
Test Data Synthesis	23
Database Subsetting	23
Master Data Management (MDM)	24
Enterprise Metadata Management (EMM)	24
<b>Analytics</b> 	<b>25</b>
Data Preparation for Other Platforms	26
Embedded BI & Reporting	26
Embedded Analytics & BIRT Visualization	28
Clickstream Analytics	28
Customer Segmentation	28
<b>Voracity Curation (Data Lifecycle Management)</b>	<b>29</b>

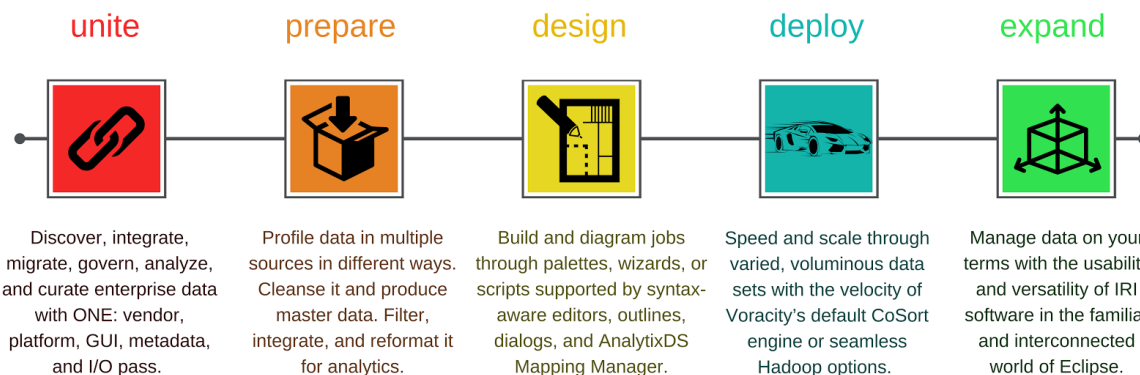
<b>Technical Specifications</b>	<b>30</b>
<b>Installation</b>	<b>30</b>
<b>Invocation</b>	<b>30</b>
<b>Data Discovery (Profiling)</b>	<b>30</b>
<b>Input and Output</b>	<b>30</b>
<b>Record Selection and Grouping</b>	<b>31</b>
<b>Sort Key Processing</b>	<b>31</b>
<b>Record Reformatting</b>	<b>31</b>
<b>Data Cleansing, Reformatting, Protection and Validation (Governance)</b>	<b>32</b>
<b>Record Summarization</b>	<b>32</b>
<b>Data Discovery (Analytics)</b>	<b>32</b>
<b>Metadata Controls</b>	<b>33</b>
<b>Resource Controls</b>	<b>33</b>
<b>Ease of Use</b>	<b>33</b>
<b>Licensing Information</b>	<b>34</b>
<b>Professional Services</b>	<b>34</b>

# Introduction

To speed insight from variable data sizes, formats, and streams, costly data integration or virtualization suites are often combined with huge servers, or complex Hadoop, NoSQL, and analytic platforms. Meanwhile, specialized data acquisition and classification, cleansing and validation, data masking and testing, MDM, and other disparate tools are usually cobbled together to close data governance gaps.

What IRI Voracity® provides instead is a modern, integrated development environment for data-driven IT departments and digital business users who need faster information and solution delivery, proven (but simpler) big data technologies, privacy law compliance, and long-term affordability. Voracity rapidly profiles, cleanses and harmonizes diverse data, protects it, reports on or wrangles it, and supports tracking data through time.

## Voracity Speeds Data's Time to Value



Built on Eclipse™ and powered by IRI CoSort® or seamlessly interchangeable Hadoop® engines, Voracity performs, combines, and speeds the discovery, integration, migration, governance, analytics, and testing of structured data, and to a growing extent, semi- and unstructured “dark data” sources.

This booklet highlights many of the feature-functions of Voracity, and shows how big data, ETL and specialty software users can *speed* their mission-critical data processing operations with Voracity components, or *consolidate* and *simplify* them by switching to Voracity. [Learn why Voracity is better.](#)

## Sample Uses

Following are typical or planned Voracity applications:



Big data CDI, CDR, clickstream and IOT reports. Analyze trends, predict, and promote.



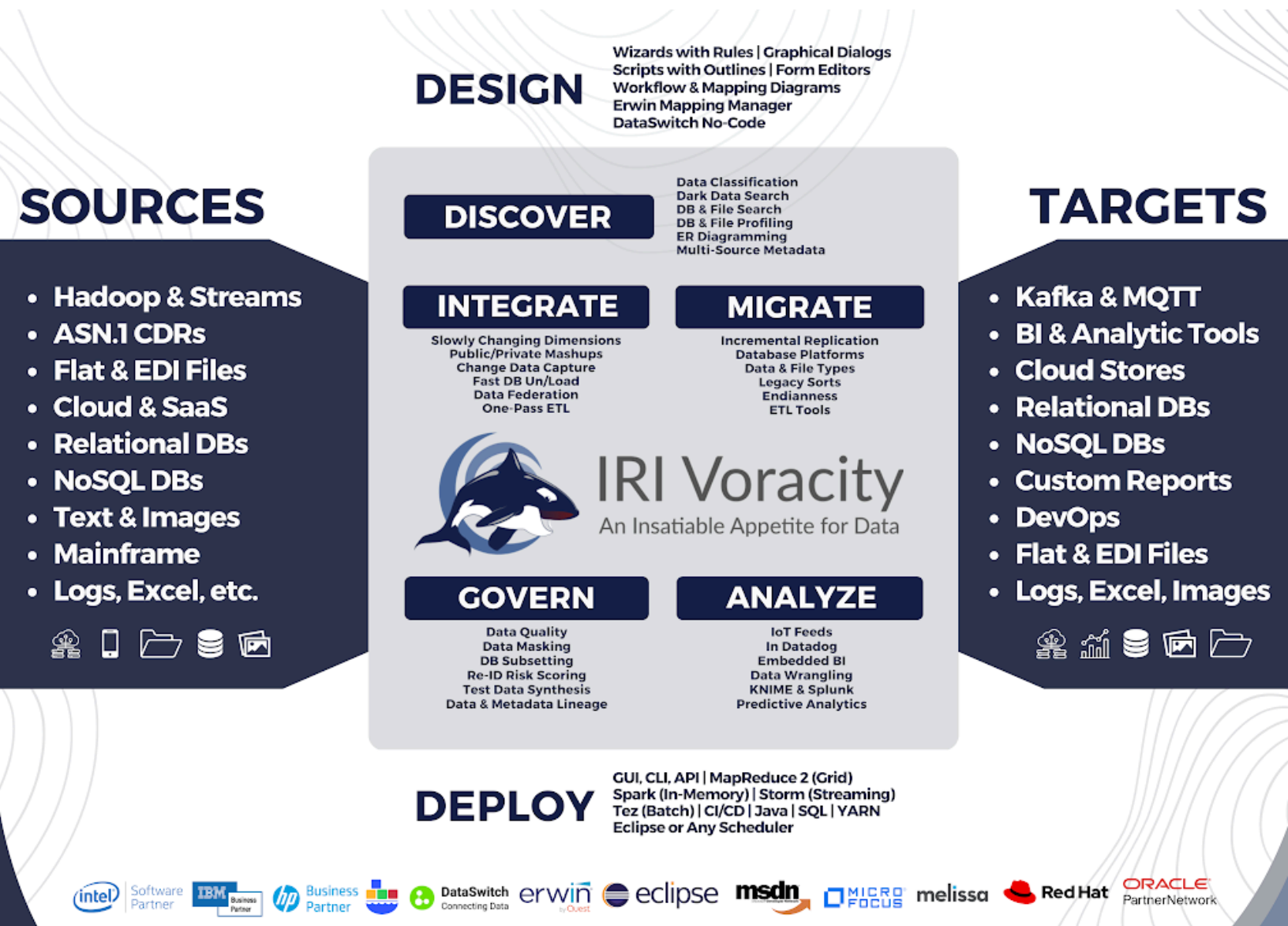
Speed, or replatform the mappings created for, slower, costlier ETL and wrangling tools.



Find, cleanse, and comply. Capture changes, fix errors, ‘master’ data, and mask PII.

# Voracity Operations

Voracity data sources, targets, and capabilities (in many cases combinable), are summarized below:



Voracity use cases include:

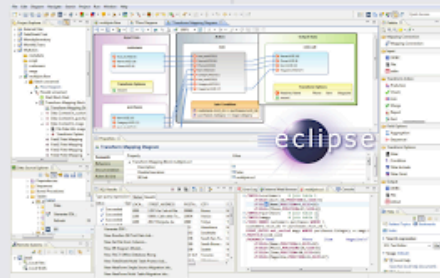
- data classification and search/discovery
- database (DB) ERD and flat-file profiles, plus “dark data” location reports/charts
- data integration (ETL) and federation
- data cleansing and enrichment (quality)
- data/file type, and DB conversion
- unload, reorg, and load acceleration
- database subsetting and replication
- un/pivoting (transposing) row data
- capturing and reporting on change data
- delta, trend, and summary reporting
- feeding KNIME and Splunk analytics
- data wrangling for 3-party BI tools
- PAN/PHI/PII/PI/CSI data masking
- test data synthesis and subsetting
- master data management (MDM)
- enterprise master data management
- legacy sort and ETL tool replacement
- building or testing Data Vault models
- updating slowly changing dimensions

The reason why Voracity platform users can address so many challenges is because of the versatility of the “SortCL” data definition 4GL and manipulation program central to Voracity’s underlying [IRI CoSort](#) data processing product. Many advanced, fit-for-purpose SortCL-compatible spin-off products – including IRI NextForm, IRI Fieldshield, and IRI RowGen – are included and/or supported in Voracity:



## IRI Data Manager Suite

## IRI Data Protector Suite



### Consolidate tools and tasks to process, protect, prototype, present

- Discover, define, and manage data in legacy and new sources
- Combine data integration, migration, governance, and analytics
- Use IRI Ripcurrent to replicate or mask changed data in real-time
- Leverage the familiarity of Eclipse and the power of CoSort

© 2024 Innovative-Routines International (IRI), Inc. All Rights Reserved. CoSort, Voracity, NextForm, DarkShield, FieldShield, CellShield, Ripcurrent and RowGen are trademarks or registered trademarks of IRI. FACT is a trademark of DataStreams Corp. All other product, brand, or company names may be (registered) trademarks of their respective holders.

### Static and dynamic masking of structured data sources

- Search, profile, and classify sensitive data in DBs and files
- Encrypt, hash, redact, pseudonymize, randomize, tokenize
- Apply cross-table rules to save time and referential integrity
- Score re-ID risk and audit your jobs to verify compliance

### Discover and de-identify PAN/PHI/PII in Excel spreadsheets

- Define or use patterns to search for sensitive data
- Locate, report, and open all found ranges in the LAN
- Click to encrypt, mask, or pseudonymize data directly
- Auto-log protections to verify privacy law compliance



### Discover, deliver, and delete sensitive information everywhere

- Find PII in LAN and cloud sources using multiple methods
- Simultaneously de-identify, remove, or report those values
- Mask text, MS, PDF, Parquet & image files + LOBs & NoSQL
- Comply with the right to erasure, portability, or rectification



### Leverage expert data privacy engineers to find and mask PII

- Avoid learning curves, software expenses and staff diversion
- Reduce risk by agreement, monitored VPN, or secure cloud
- Use operational logs for reporting and compliance audits
- Select from competitive hourly, daily or project rates

### Speed or replace legacy sorts, batch/ETL/SQL transforms

- Filter, join, aggregate, pivot, cleanse, lookup, calc, etc.
- Map, migrate, federate, and replicate data from 150 sources
- Segment data, capture changes, report details / summaries
- Analyze changing dimensions, support complex transforms



### Speed RDBMS unloads for archival, migration, reorg, and ETL

- Extract tables to flat files in parallel using SQL queries
- Convert and re-format to change data types and layouts
- Create the data definitions for IRI software and DB loads
- Pipe to CoSort and DB loaders for faster reorg and ETL



### Unlock data and move between apps, DBs, and platforms

- Convert, federate, remap, and replicate legacy data
- Migrate data between databases and create new tables
- Change file formats, data types, and endian conditions
- Find, extract, and structure data in unstructured sources



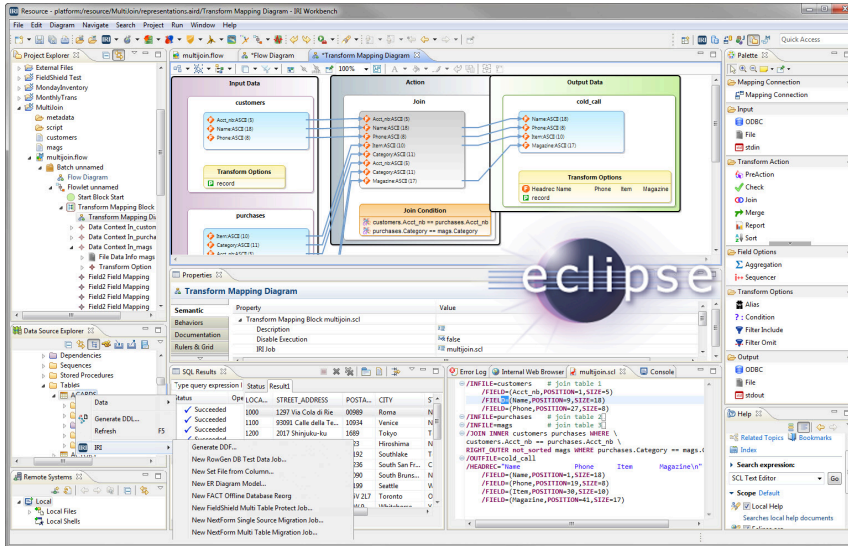
### Prototype DBs and ETL, stress-test, outsource, benchmark

- Use real data models and formats, not production data
- Combine generation and selection, create new formats
- Preserve referential integrity and frequency distributions
- Feed test DBs, files, reports, and DevOps simultaneously

## Voracity Job Development Options

To support the work styles of a diverse user base, the [Voracity GUI](#) (IRI Workbench) and its underlying metadata model supports several compatible, largely re-entrant job creation and modification methods:

1. fit-for-purpose new job creation (script generation) wizards
2. state-of-the-art visual workflow and mapping diagram palettes
3. color-coded, syntax-aware script editors with dynamic outlines
4. help-enabled graphical dialogs and form editors to modify parameters
5. [DataSwitch](#) or erwin Mapping Manager (EMM) spreadsheet-based, API-driven mapping models
6. GulfStream SDK, a Java API for SortCL scripting and Workbench workflow metadata



Voracity jobs are ultimately specified in standalone scripts created in the GUI or elsewhere.

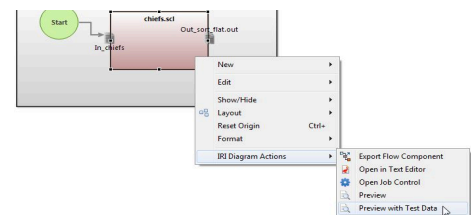
Most data manipulations are specified in 4GL scripts in the “SortCL” syntax of IRI CoSort, and encapsulated in larger XML workflow files containing SortCL and/or other job script syntax. IRI Workbench supports all IRI job scripts and metadata, SQL, 3GLs, plus CI/CD (DevOps) and other Eclipse plug-ins.

More specifically, Voracity workflows and diagrams illustrate, model, and support the syntax of:

1. IRI FACT (Fast Extract) .ini configuration files
2. IRI Sort Control Language (SortCL) programs -- and the data definition file (DDF) metadata -- used in IRI CoSort® transformation (for Voracity ETL) and reporting, IRI FieldShield® data masking, IRI NextForm® data/DB migration, and IRI RowGen® test data creation scripts
3. Command-line (system) calls, and shell (batch) scripts
4. SQL (stored) procedures and Java programs
5. Other apps supported in Eclipse that process data, or interact with the above jobs like Jenkins

Voracity users can also create and re-use mapping, masking, or (test) data generation rules, with or without associated data class libraries. Visual SQL support, text editors, and local/remote shells are included. Free plug-ins for COBOL, C/C++, Hive, Impala, Java, Perl, Python, and R are available.

The Voracity IDE also offers multiple job debugging and deployment options. The workflow palette allows you to specify one or more tasks (flowlets) within a larger project (flow), edit properties, and preview mapping targets using either a subset of production data, or auto-generated test data.



## Voracity Job Deployment Options

- |  |   |
|--|---|
| 1) Command line (any shell) or batch script  | 8) Web service (e.g. node.js) calls   |
| 2) Built-in (IRI Workbench) <a href="#">task scheduler</a>                             | 9) 3GL (Java, C/C++, COBOL) calls   |
| 3) Third-party schedulers (cron, <a href="#">UAC</a> , etc.)                           | 10) <a href="#">GitLab</a> , <a href="#">Azure</a> DevOps, <a href="#">AWS</a>            |
| 4) <a href="#">Voracity node for KNIME</a>   | CodePipeline, or <a href="#">Jenkins</a> CI/CD tasks                                      |
| 5) <a href="#">Voracity app for Splunk</a>   | 11) <a href="#">Actifio</a> , <a href="#">Commvault</a> , or <a href="#">Windocks</a> DB  |
| 6) <a href="#">DataSwitch</a> no-code data platform                                    | cloning operations  |
| 7) Hadoop MR2, Spark, Spark Stream, Storm or Tez via “ <a href="#">VGrid</a> ” gateway | 12) FieldShield or RowGen jobs run from <a href="#">Value Labs</a> or Cigniti TDM portals |

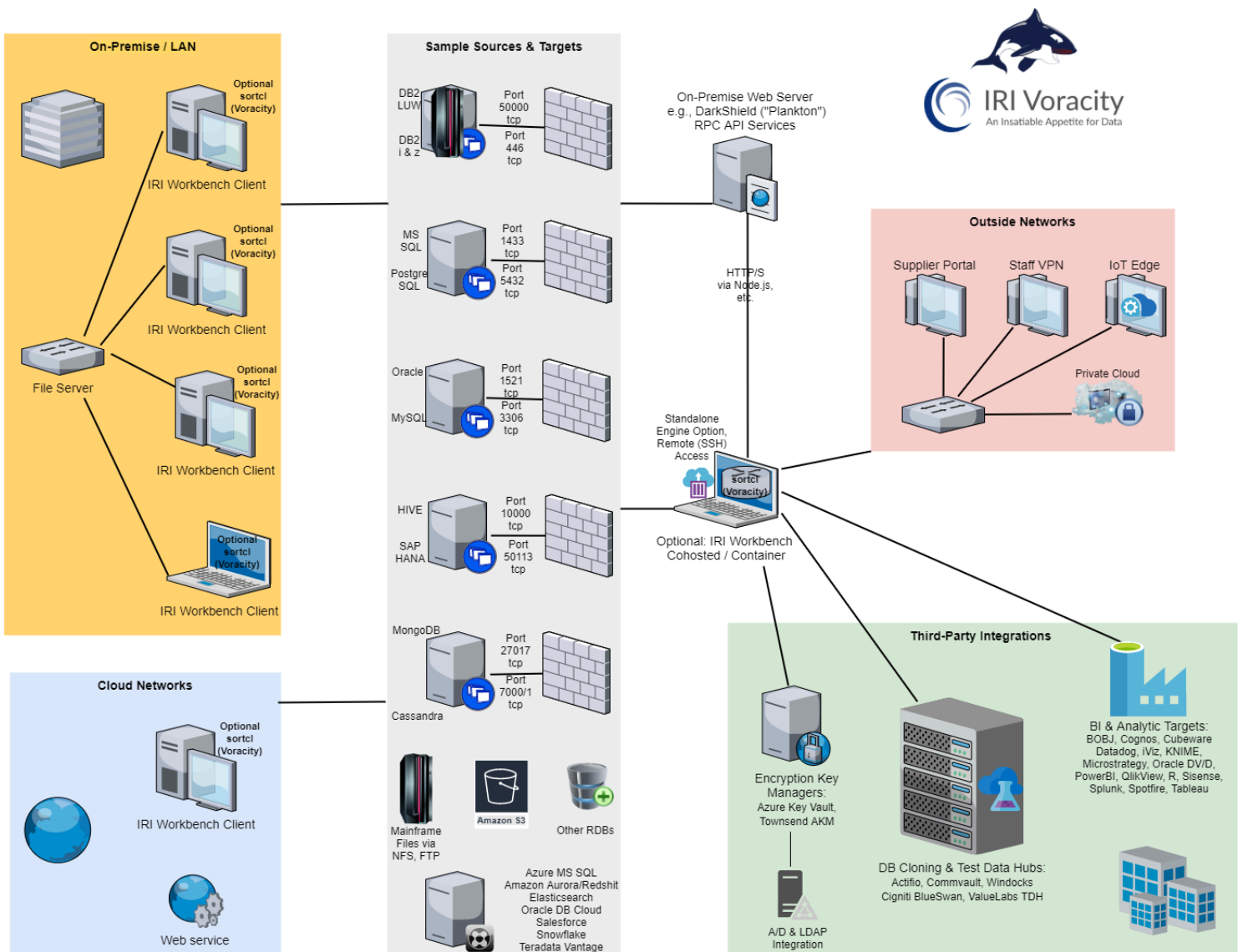


# Voracity Resource Requirements

For x86 systems, a minimum configuration for Workbench would be 4GB of RAM and 10GB of free disk space, after the installation of any VMs, DBs, etc. However, 6GB and up works best for each system to accommodate multiple database connections and table parsing for metadata and job definition. For schemas with hundreds of tables to enumerate, as much as 64GB of RAM could be appropriate for the Workbench machine(s) where RDB-related jobs are built.

IRI also recommends where possible the co-location of the licensed back-end (SortCL executable) on or within close network proximity to database source or target servers for performance reasons, particularly if there are known network bottlenecks. Data maps, masks, munges, and mines essentially at movement speed, so consider network and I/O resources.

## IRI Voracity Communication & Networking Architecture



*The default Voracity stack uses both the front-end IRI Workbench graphical IDE for client-side design of data-driven jobs defined in portable CoSort SortCL scripts. The back-end SortCL engine which runs these jobs is the default, C-language executable supporting Windows, Linux and Unix systems ranging from a Raspberry Pi to a z/Series mainframe. Many of the same scripts also run interchangeably in Hadoop.*

# Data Discovery



# Data Discovery (Profiling)



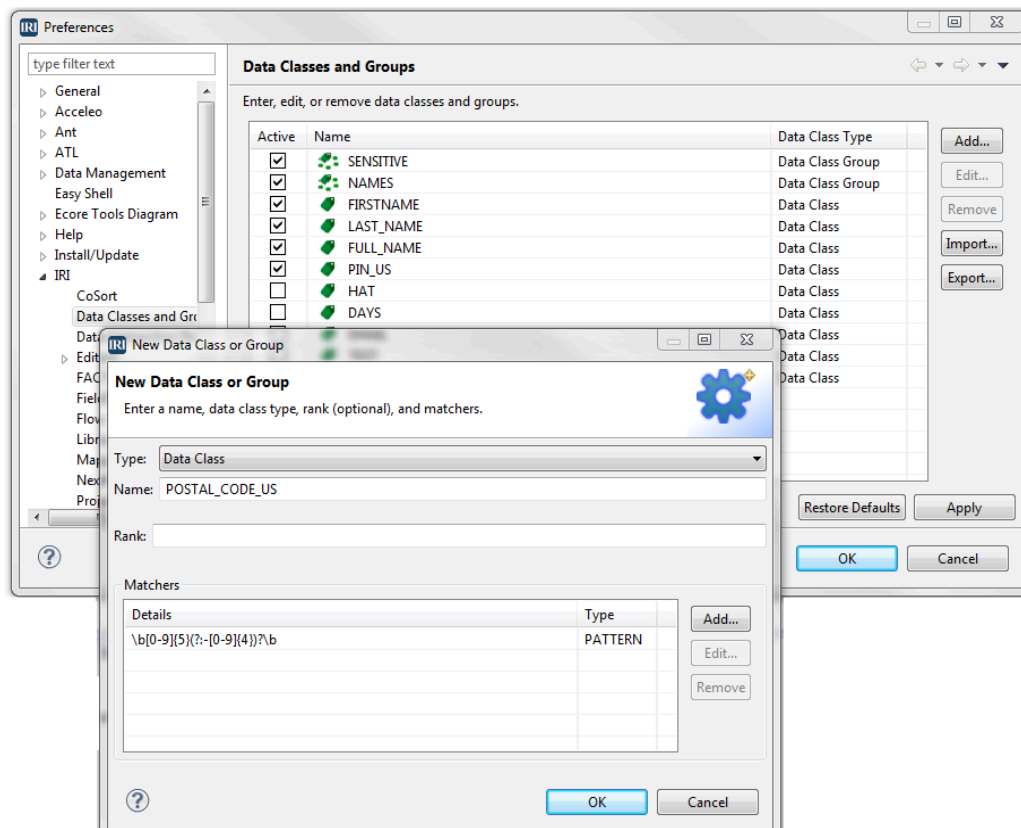
With more data being culled from more aspects of business today, awareness of its content and nature is vital to ensuring its quality, quantity, and security. Voracity [data discovery tools](#):

- find and group data into class libraries for transformation, migration, masking and test data rules
- reveal and validate relationships between entities
- produce statistical information on data
- perform pattern-matching, fuzzy-matching, machine-learned NLP NER, and dictionary searches
- expose forensic metadata through logs, and share them with SIEM tools [like Splunk](#)
- discover or auto-define source metadata

across disparate [data sources](#), including DB tables, flat files, and dark data (unstructured) documents. Voracity allows users to examine the structure and completeness of database data, and validate that the proper data is being stored in the right places. The discovery process supports data cleansing, integration, masking, reformatting, and reporting in Voracity.

## Data Classification

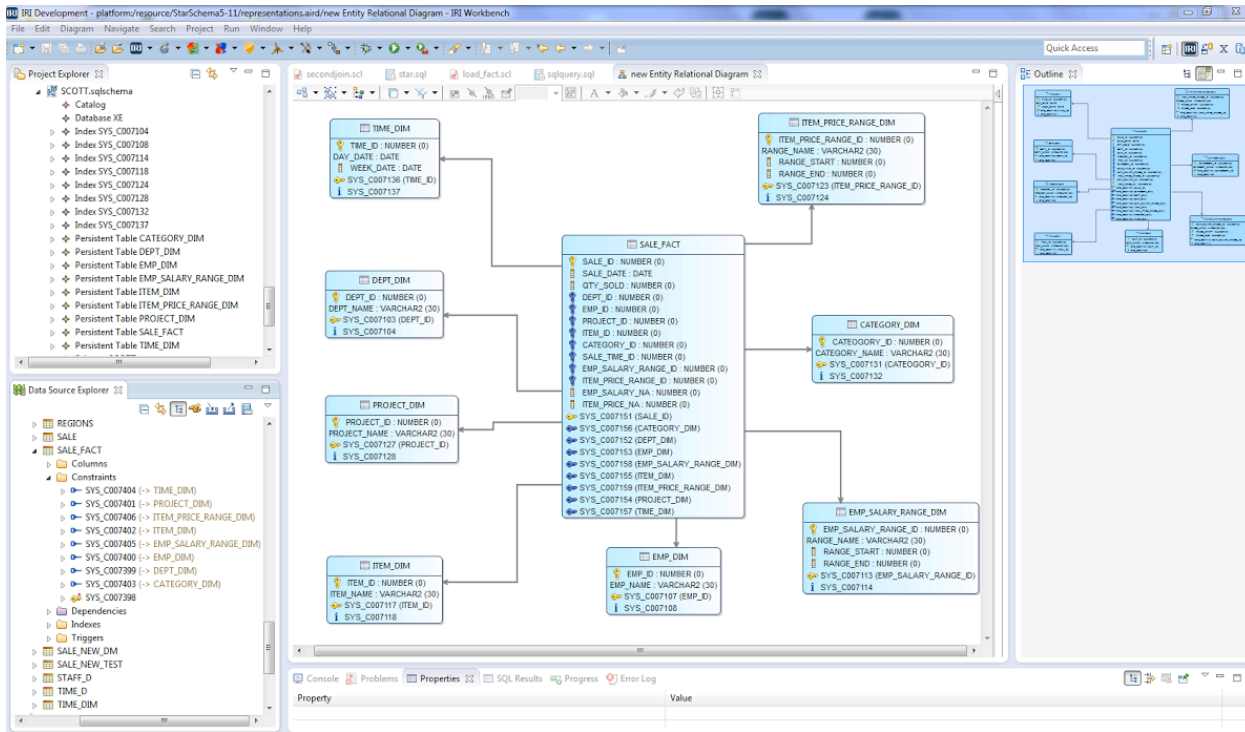
Define and manage enterprise-wide data [class libraries](#), and use them to apply [field rules](#) for data transformation and protection across multiple data sources at once.





# E-R Diagrams

The Entity-Relationship Diagram (ERD), or model, shows database tables (entities) and how they are linked through primary and foreign keys (relationships) to each other. Use Voracity to analyze the structure of, and relationships between, tables you [select graphically](#) in any connected database.



# Database Profiling

**Column Selection**

Select a table and options to specify what you would like generated in your database profile.

Connection profile: Twister Oracle

SCOTT.AIRPORT  
SCOTT.ANEW  
SCOTT.ANULLS  
SCOTT.AUGUST  
SCOTT.AUGUST1  
SCOTT.BONUS  
SCOTT.CARS  
SCOTT.CARS\_ORACLE

Perform full profile on all columns in all tables.

Columns:

Options	ID	NAME	MANU...	PRODU...	HO...
<input type="checkbox"/> [All]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> [Basic]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Row Count	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...
<input type="checkbox"/> Distinct Cou...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Duplicate C...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Null Count	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Empty Count	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Value Count	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> First Value	<input checked="" type="checkbox"/> Ch...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Last Value	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> [length]	<input type="checkbox"/>	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...
<input type="checkbox"/> Min. Length	<input type="checkbox"/>	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...
<input type="checkbox"/> Max. Length	<input type="checkbox"/>	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...
<input type="checkbox"/> Avg. Length	<input type="checkbox"/>	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...
<input type="checkbox"/> Med. Length	<input type="checkbox"/>	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...	<input checked="" type="checkbox"/> Ch...
<input checked="" type="checkbox"/> [Expression ...]	<input checked="" type="checkbox"/> Ch...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> [Check Refe...]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Expression Search**

Specify one or more expressions to report on the values in selected table columns

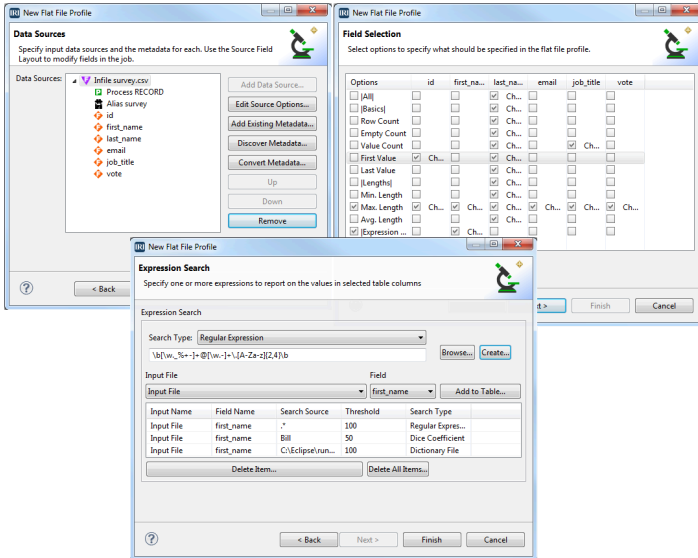
Expression Search

Search Type: Regular Expression

Search: \b(?:\s\*Visa>4[0-9]{12}(?:[0-9]{3})?)|(?:\s\*MasterCard>5[1-5]{0-9}[14])?</p></div>

Get statistics, check referential integrity, and search for string-, pattern-, fuzzy-matching, and "set file" lookup values within [any database](#) you connect to via JDBC in Eclipse.

11

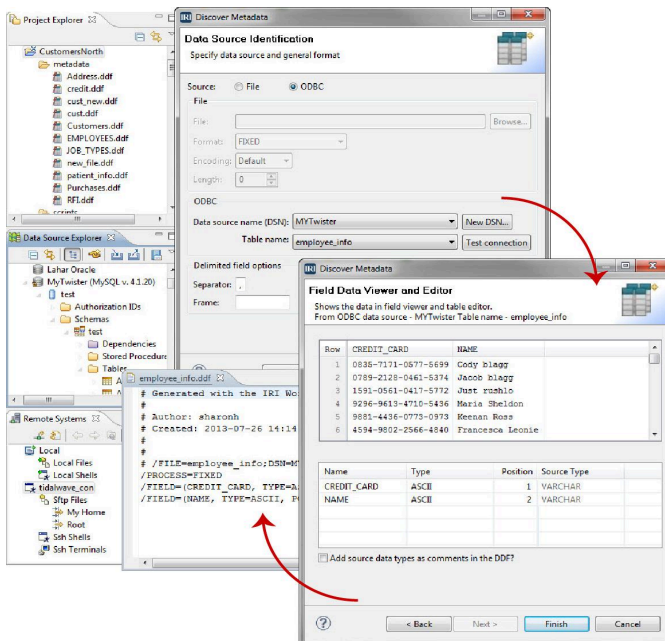
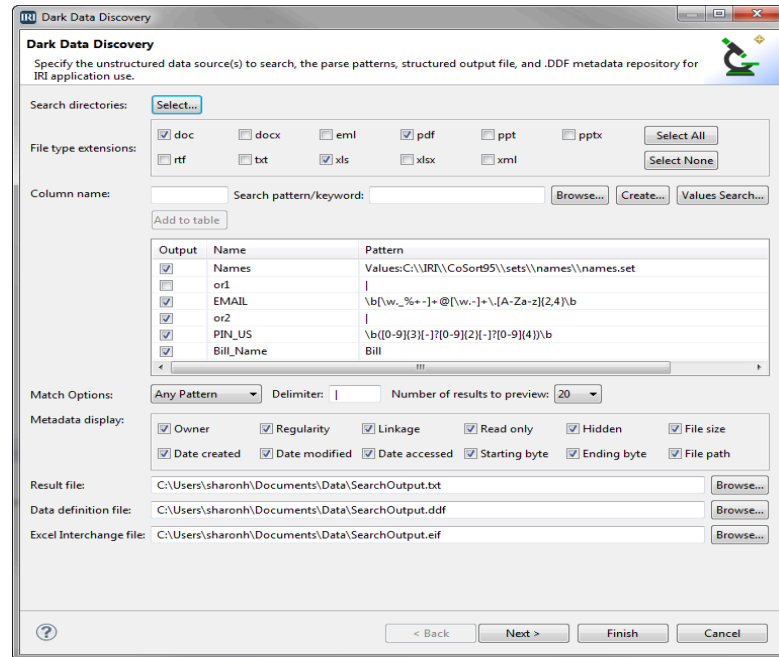


## Flat-File Profiling

Get statistics, and search for string-, pattern-, fuzzy-matching, and lookup value matches in **flat files**; i.e. COBOL and line, record or variable sequential, delimited (including CSV) and fixed text, CLF and ELF web log, Excel, LDIF, and consistent JSON and XML file formats.

## Unstructured (Dark) Data Discovery

Find values in PDF and MF Office documents, unstructured text (log, chat, email), and image files, and semi-structured database sources on premise or in the cloud. Define patterns, path filters, NER models, or string matches. Extract the values and forensic metadata from source files into flat files. Use that data and the flat-file DDF metadata in Voracity data integration, replication, protection, reporting, and other jobs. Mask the data automatically using the built-in IRI DarkShield product ... so you can discover, deliver, and de-identify PII subject to the GDPR, CCPA, etc.



## Structured Metadata Discovery

Connect to flat files and relational databases (RDBs) to review and define their layouts in the data definition format (DDF) files Voracity (and all IRI software) uses in data manipulation and reporting jobs. Parse ODBC, Excel, MongoDB, JSON, XML and other and delimited file sources automatically, and fixed position files manually. Automatically create DDFs from COBOL copybooks and DB loader files, and CSV, DDL and DB loader files from DDFs.

# Data Integration



# Data Integration



The challenge of distilling information from data only grows with its volume and variety. Voracity provides a full set of data integration capabilities that are faster and simpler than legacy tools. It is also more affordable to operate, and easier to adapt to changing data sets and analytic needs.

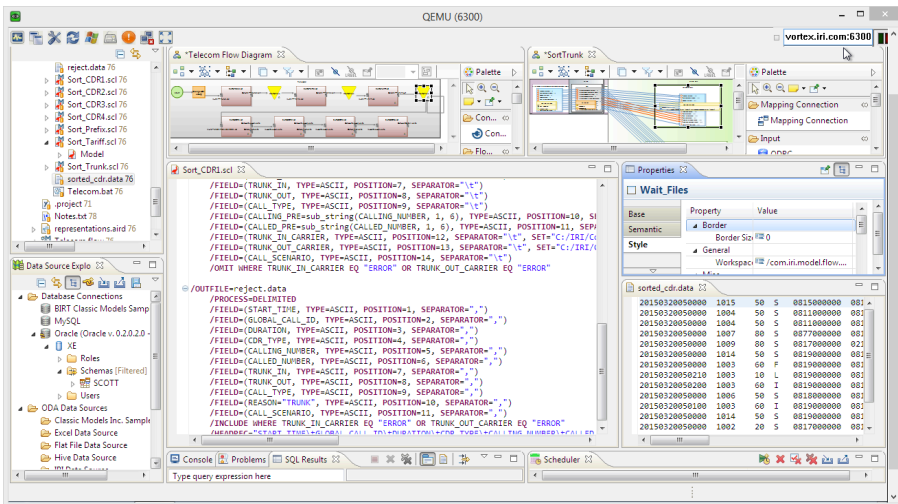
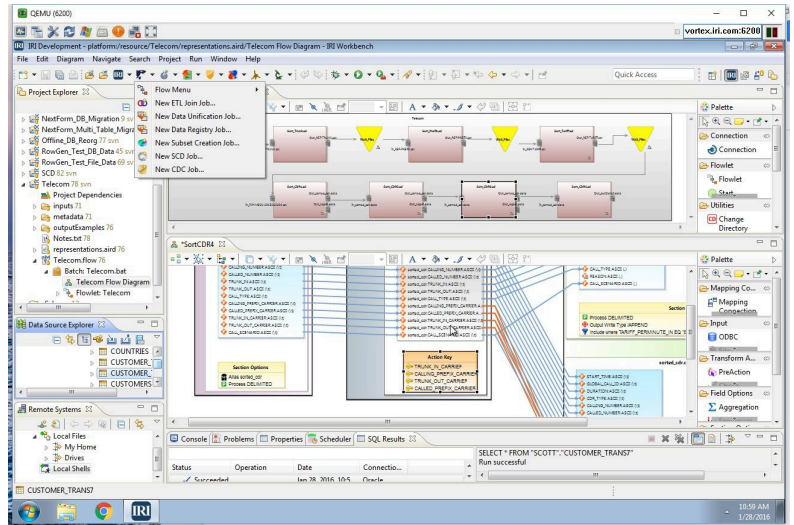
## Single-Pass ETL

Voracity integrates data in big data warehouses and data lakes, as it optimizes and combines:

1. Extraction, via IRI **FACT** (F**A**st extra**CT**) -- a parallel unload utility for VLDB tables
2. Transformation, via IRI **CoSort** or Hadoop MapReduce 2, Spark, Spark Stream, Storm, or Tez
3. Loading, via native DB load utilities that benefit from piped-in, pre-CoSorted data

in the same workflow and I/O pass with data streaming in memory (e.g. pipes) between steps. And in that single pipeline, Voracity can simultaneously:

- filter, sort, join, and aggregate
- cleanse and de-duplicate
- conditionally filter and segment
- convert file, data and endian types
- remap and mask fields
- perform lookup and pivot transforms
- cross-calculate and rank
- encrypt, redact, hash, etc.
- label and silo changed data
- report deltas, stats, and summaries
- bulk load with pre-sorted data
- replicate and federate data
- feed data into KNIME, Splunk, etc.



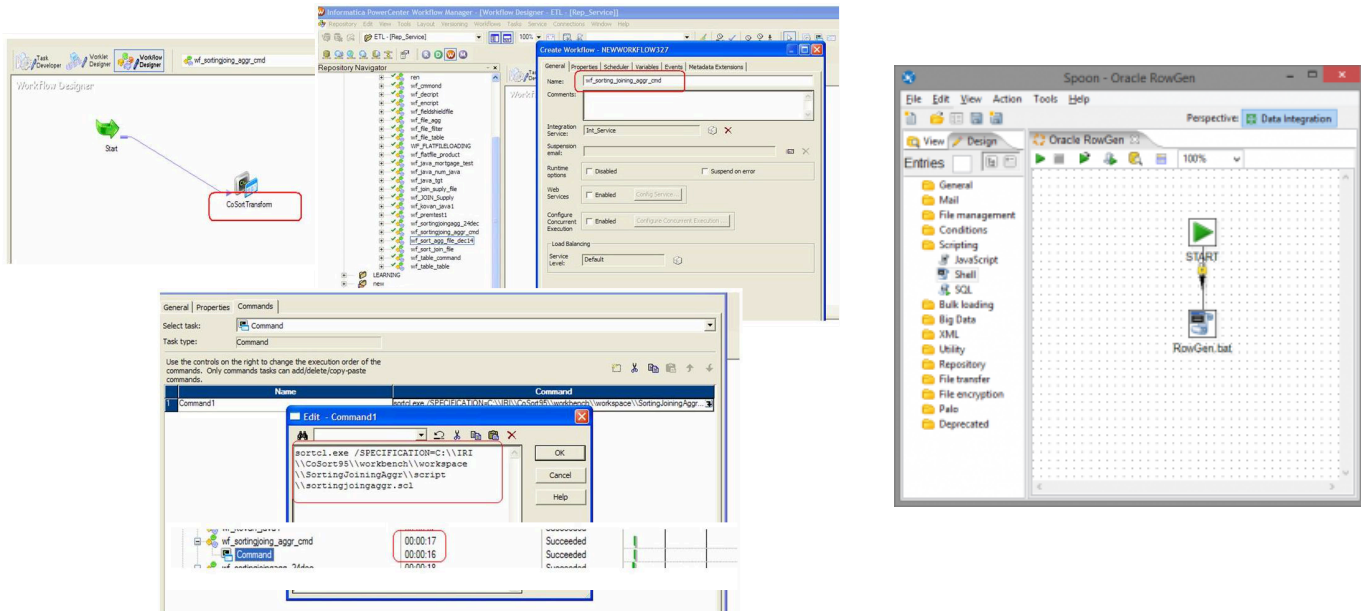
**Voracity Metadata:**  
*Self-Documenting. Re-Entrant.  
Easy to Learn and Manage*

In Voracity the: 1) data layout and job specification metadata, 2) data discovery and acquisition steps that support them, 3) key dependencies defined between them; and, 4) the target reports and hand-offs they produce, are all specified in a 4GL that you can edit or manage graphically.

## Legacy ETL Tool Acceleration & Augmentation



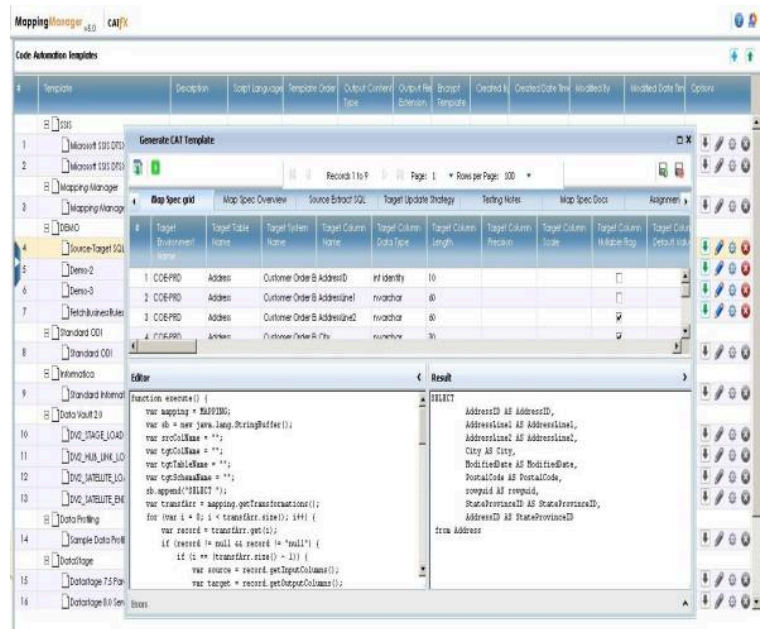
Voracity components like IRI FACT can speed [extracts](#) for other ETL tools, while its IRI CoSort engine can [optimize transforms](#) and [bulk loads](#). Built-in IRI FieldShield functionality can [mask data](#) ingested by other ETL tools, while its RowGen pieces can [generate perfect test data](#) for them.



## Legacy ETL Tool Migration

Either the [DataSwitch](#) (no-code, AI-enabled data engineering platform), or erwin by Quest [Mapping Manager](#), can automate the conversion of jobs and metadata between ETL platforms. Their frameworks and services can re-platform ETL mappings to faster, simpler Voracity ETL jobs from:

- Ab Initio
- Actian/Pervasive Data Integrator
- IBM InfoSphere DataStage
- Informatica PowerCenter
- Microsoft SQL Server SSIS
- Oracle Data Integrator
- Oracle Warehouse Builder
- Pentaho
- SQL procedures
- SyncSort DMx
- Talend





## Change Data Capture



Change data capture (CDC) identifies and tracks data that has changed, so that actions can be taken. CDC in Voracity is mostly data-centric, vs. log-centric. This precludes the need for log sniffers, DB-specific triggers and other complexities, while supporting:

- multiple sources, not just one DB
- insert, delete, and update segmentation
- calculation and reports on update values
- simultaneous transformations
- new data types, offsets, and target formats
- PII masking via encryption, redaction, etc.
- detail and summary report layouts
- refresh tables with real-time updates
- bulk load pre-sorts into DB utilities
- archive, replication or hand-off files

Regardless of the target(s), this approach removes a major workload from the DBMS, because it need not rely on triggers to update the tables.

## Slowly Changing Dimensions (SCD)

Voracity processes SCD type 1, 2, 3, 4 and 6 updates in one convenient job creation wizard to facilitate the refreshment of dimensional data.

Voracity also allows you to query and report on discrete values in a master source based on changing information, like date and time. This reporting capability supports:

- very fast lookup performance
- searches on any strictly increasing value
- complex, multi-level search criteria
- simple job script maintenance and sharing
- fast application of, and ETL on, new values
- support for built-in comments
- the need for DB transformation, reorgs, etc.



## Data Federation

Data federation bypasses the normal ETL overhead of physical data consolidation, replication, and relational database population. Voracity can virtualize, rather than persist, disparate data source integration to produce immediate, fit-for-purpose informational views. To produce ad hoc information from siloed sources, simply specify the input sources and console (stdout)- or procedure-directed target layouts as part of any data mapping operation.

The screenshot shows the IRI Workbench interface. The main window displays a 'Stockjoin Mapping Diagram' with three main sections: 'Input Data', 'Action', and 'Output Data'. The 'Input Data' section includes a 'Join' action with a 'Join Condition' box. The 'Output Data' section shows a 'stdout' target. The console window at the bottom displays a SQL query and its output table.

```
28 #/JOIN FULL OUTER NOT_SORTED nyse-a NOT_SORTED buys WHERE nyse-a.
29
30 ##### Output File 1 - 2D BI Report with Selection #####
31 #/OUTFILE=..\Targets\TradingA # summary record format
32 #/OUTFILE=stdout
33 #/HEADREC=
34 #/FIELD=(New_balance,POS=50,SIZE=14,PRECISION=2,currency)
35 #/SUM New_balance from (nyse-a.LastTrade * buys.Shares) # ex
36
37 #/OUTFILE=..\Targets\TradingA # detail record format
38 #/OUTFILE=stdout
39 #/HEADREC="Client Symbol Shares LastTrad
40 #/FIELD=(buys.Client,POS=1,SIZE=17)
41 #/FIELD=(buys.Symbol,POS=20,SIZE=5)
42 #/FIELD=(nyse-a.Symbol,POS=28,SIZE=5)
43 #/FIELD=(buys.Shares,POS=35,SIZE=5)
44 #/FIELD=(nyse-a.LastTrade,POS=45,SIZE=5,PRECISION=2,NUMERIC)
45 #/FIELD=(product,POS=54,NUMERIC, \
```

Client	Symbol	Shares	LastTrade	Shares*LT	Ln.
	ABB		12.55		1
	ABM		16.44		2
Jack Welch	ABN	2000	27.47	54940.00	3
Lakshmi Mittal	ABT	825	47.25	38981.25	4
Robert Kiyosaki	ABY	9000	2.61	23490.00	5
Lisa Mangino	ADS	855			6
Michael Bloomberg	AGE	1500	52.81	79215.00	7
	AIC		4.84		8
Jeff Bezos	AMR	3250			9

Rather than creating new middleware, a federated database, or application layer, data federation occurs in the same place that ETL and BI do: in the IRI Workbench. See [this series](#) of articles on *Voracity as a Production Analytics Platform*, authored by Dr. Barry Devlin of 9Sight Consulting.

## Public/Private Mashups

Companies collect transactional data in the course of business, while government agencies and NGOs collect large volumes of demographic, economic, scientific activity data that can be leveraged and combined with internal data to create new insights. For example, in a customer service scenario, public weather and news data might be [combined](#) with customer service alerts and locations to build a customer service portal. Voracity can connect to differently-formatted, static or streaming data sources, join related elements, provide near-real-time reports, and feed OLAP and [Splunk targets automatically](#).

# Data Migration





# Data Migration

Data migration occurs for a variety of reasons, including server or storage equipment replacements, maintenance or upgrades, application migration, website consolidation, and data center relocation. To help move data between systems, Voracity can:

- Reveal the location, layout, and relationships of data in DBs, files, and documents (see Data Discovery)
- Remap data types, record layouts, file formats, and endianness
- Convert column data, layouts, and constraints between DBs
- Migrate schema
- Replicate, or copy, data from one or more sources to similar or different targets
- Create hand-offs, persistent reports, or federated views (see Analytics)

The screenshot displays the Voracity software interface with several windows open. The 'New Multi Table Migration Job' window shows configuration for a 'Data Extraction Stage' with 'Tables selected: 2' and a 'Field Modification Rules' window. A 'Summary' window provides details about the ODBC connection and rule information. A 'Single Source Migration' window is also visible. The main interface shows a 'Transform Mapping Diagram' with 'Input Data', 'Action', and 'Output Data' stages. A 'Data Source Explorer' window shows a tree view of data sources, and a 'Console' window displays a table of patient data.

Id	number	name	age	email	phone	ssn	city	state	start_date	end_date
1	664232	Adams John	68	adams@gmail...	646-834-9956	770-75-3849	Melbourne	FL	1997-06-11	2000-10-19
2	68346890	Jefferson Tho...	22	jefferson@Th...	321-890-8293	345-78-3485	Goodland	Kansas	1996-05-05	2007-05-29
3	833945	Madison James	44	Madison@Jam...	563-425-7822	643-38-3878	Blount	Mississippi	2010-04-01	2010-05-17
4	94967205	Monroe James	52	Monroe@Jam...	433-758-2783	123-45-4783	Rapid City	South Dakota	2003-05-20	2007-05-16
5	49002045	Jackson Andrew	90	Jackson@Andr...	143-894-4328	335-99-6655	Long Beach	New York	1970-05-12	2008-05-09
6	23628647	Adams Quincy	10	Adams@Quinc...	983-245-2994	234-87-1122	Lynn	Nebraska	2003-05-28	2009-05-14
7	5629255	Van Martin	87	Van@Martin...	684-763-7612	763-76-2348	Tulsa	Oklahoma	2000-05-17	2010-05-10
8	77890346	Harrison Henry	53	Harrison@Hen...	765-978-2457	574-57-8934	Aberdeen	South Dakota	1997-05-26	2010-05-14
9	77920263	Taylor Zachary	45	Taylor@Zacha...	423-500-3289	678-99-7474	Charleston	West Virginia	1996-05-22	2010-05-03

## Data-Type, File-Format & Database Migration

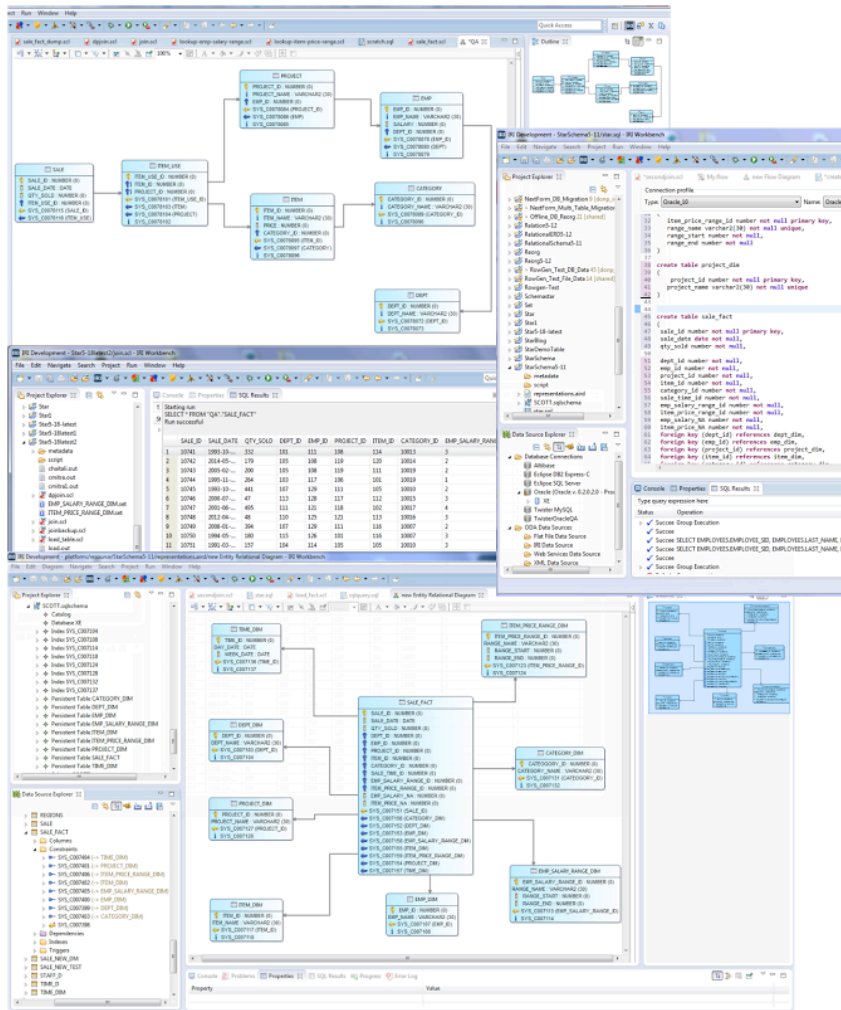
Voracity converts data types during data integration, migration, replication, federation, and reporting. Translate fields from EBCDIC to ASCII, packed decimal to numeric, American to ISO timestamp, native CJK to Unicode, IP address to whole number, etc. Convert file formats in the same situations; e.g. Micro Focus I-SAM file to text, fixed to JSON, LDIF to CSV. For database migrations, Voracity will map data and constraints from one database, and load them in bulk into another.

## Endian Migration

Voracity's ability to change file endianness allows data to process correctly on any hardware, facilitating work offloads to more available or cheaper servers. Field-level endian control facilitates data integration between heterogeneous files and DBs. Producing big and little endian targets simultaneously reduces data remediation, redundancy, and synchronization problems.



# Schema Migration



Voracity users can not only visualize their schemas, they can convert them.

Information about the structure and relationships between relational entities is parsed to produce source metadata ready for manual or wizard-driven mapping into new schemas.

This example shows the migration of a standard relational database schema into a data warehouse star schema, but other types of schema changes are possible within the same, or between different, databases.

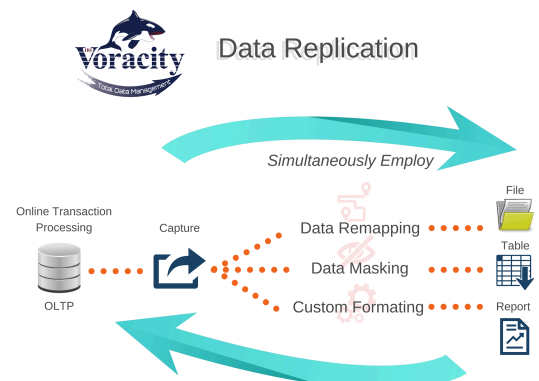
Voracity also has a fit-for-purpose [Data Vault Generator](#) wizard, to migrate relational database schema and data to Data Vault 2.0 models, or to generate and populate them with compliant Data Vault test data.

# Data Replication

Voracity users can make multiple copies of various sizes and shapes of their data sources at once, and apply a series filters, conversions, mappings, and PII maskings, too. Replicas can be designated files, tables, pipes, reports, and procedures, and even serve as new data sources in the same workflow. And, their creation can be based on event triggers.

Data replication in IRI Voracity is inherently:

- scalable in volume
- database- and platform-agnostic
- reliable in terms of data (and referential) integrity
- functionally flexible, and optionally incremental
- less expensive than specialty replication software



# Data Governance





# Data Governance

Voracity provides a simple, cohesive framework in which to govern and protect data while you: move it, transform it, cleanse it, report on it, and otherwise prepare it for analysis or applications.

Most tools that do things with data are not integrated, and their interplays add layers of complexity that can impact performance and the availability of data. They also do not provide the audit trails that help satisfy the requirements of your risk and controls framework.

The IRI Voracity data management platforms supports these data governance activities:

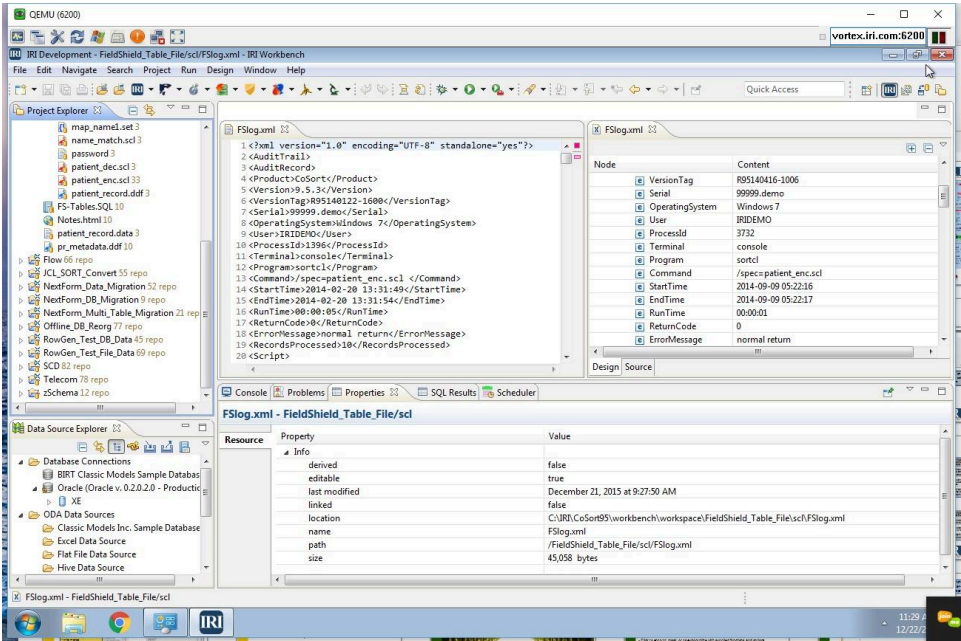
- goal, policy, and standard (rule) setting
- data profiling and metadata definition (see data discovery above)
- data quality: search, validate, cleanse, enrich
- static and dynamic data masking
- smart, synthetic test data generation
- enterprise metadata management (EMM)
- master data management (MDM)
- metadata and task lineage, sharing & security
- forensic metadata discovery and job auditing

## Business Goals, Policy, and Rules

Voracity conforms to most COBIT 5 framework recommendations by supporting the identification, recording, and enforcement of data governance objectives, policies, and procedures. For example, Voracity users can define and re-use searches of data in multiple sources based on their business needs and applicable cybersecurity (data privacy) laws.

As or after it gets found, sensitive data -- and its metadata -- can be extracted, saved, analyzed, converted, shared, or secured as needed.

The same applies to other process information Voracity applies; e.g. project and configuration information, masking and mapping rules, job configuration and scheduling, change/lineage data, and audit logs like this:





## Data Quality

Voracity can scrub and standardize named fields and other attributes of table and file data to cleanse, enrich, and standardize transaction data, or build master data repositories.

Data filtering, validation, enrichment, and reformatting logic can be [specified as rules in the IRI Workbench GUI for Voracity](#), and serialized in SortCL job scripts. Thus data quality can be enhanced as a standalone matter, or within the same job script and I/O pass along with ETL, data masking, reporting, etc. Data standardization can be part of that pass too, via field-level calls to a third-party library functions. Fit-for-purpose master data management (MDM) wizards unify 'like' data through 'fuzzy' matching.

<b>DQ Capability</b>	<b>Options</b>
Profile & Assess	<a href="#">Discover</a> and analyze sources and metadata in fit-for-purpose data profiling wizards in the <a href="#">IRI Workbench</a> (Eclipse GUI).
Bulk Filter	Remove unwanted rows, columns, and duplicate records with equal sort keys. Use selection logic to omit or save bad values.
Validate	Use field-level 'iscompare' functions to isolate null values and incorrect data formats. Use outer joins to silo source values that do not conform to master (reference) data sets. Use data formatting templates and their date validation capabilities to check the correctness of input days and dates.
Unify (MDM)	Use the consolidation-style <a href="#">data unification</a> wizard to find and assess data similarities, and remove redundancies. Bucket the remaining master data values in new files or tables. Use the registry-style wizard to direct new master data into its sources.
Replace	Specify one-to-one replacement via PCRE search functions, or create multiple values in sets used for many-to-one mappings.
De-duplicate	Eliminate duplicate rows with equal keys.
Cleanse	Specify custom, complex include/omit conditions based on data values or business (SQL query) logic.
Enrich	Combine, sort, join, aggregate, lookups and segment data from multiple sources to enhance row and column detail. Create new data forms and layouts through conversions, calculations and expressions. Enhance layouts by remapping and templating (composite formats). Build more or new test data.
Standardize	Run field-level calls to Trillium or Melissa Data (e.g. address) standardization APIs to integrate their DQ at runtime.
Generate	Create good and bad data, including <a href="#">realistic</a> values and formats, valid days and dates, national ID #s, master data, etc.





# Data Masking & Re-ID Risk Determination

Voracity users can leverage all of the data discovery, masking and reporting capabilities of [IRI FieldShield](#), [IRI CellShield EE](#), and [IRI DarkShield](#), to classify, find, and protect personally identifiable information (PII) in DBs, spreadsheets, and files -- structured, semi-structured, and unstructured.

The masking functions in FieldShield (some of which are also in CellShield EE and DarkShield) are:

<i>blurring</i>	<i>encryption</i>	<i>hashing</i>	<i>de-ID</i>	<i>pseudonymization</i>
<i>bucketing</i>	<i>encoding</i>	<i>filtering</i>	<i>redaction</i>	<i>randomization</i>
<i>omission</i>	<i>conversion</i>	<i>string functions</i>	<i>expressions</i>	<i>tokenization</i>

These functions nullify the effect of any data breach, and allow you to:

- choose the mask for each identifier to comply with privacy laws and business needs
- save time, money, and inconvenience by not protecting non-sensitive data
- maintain data realism and referential integrity
- improve efficiency by combining data protection with data transformation and reporting
- send compliant data to applications, reports, databases, the cloud, and BI tools
- implement data loss prevention (DLP) programs properly, and without undue complexity
- verify compliance with full, query-ready audit logs of the protection jobs

A built-in [risk-scoring wizard](#) statistically measures, and graphically reports on, the probabilities that masked data sets can still be used to re-identify individuals on the basis of remaining quasi-identifiers.

The image displays several overlapping screenshots of the FieldShield software interface:

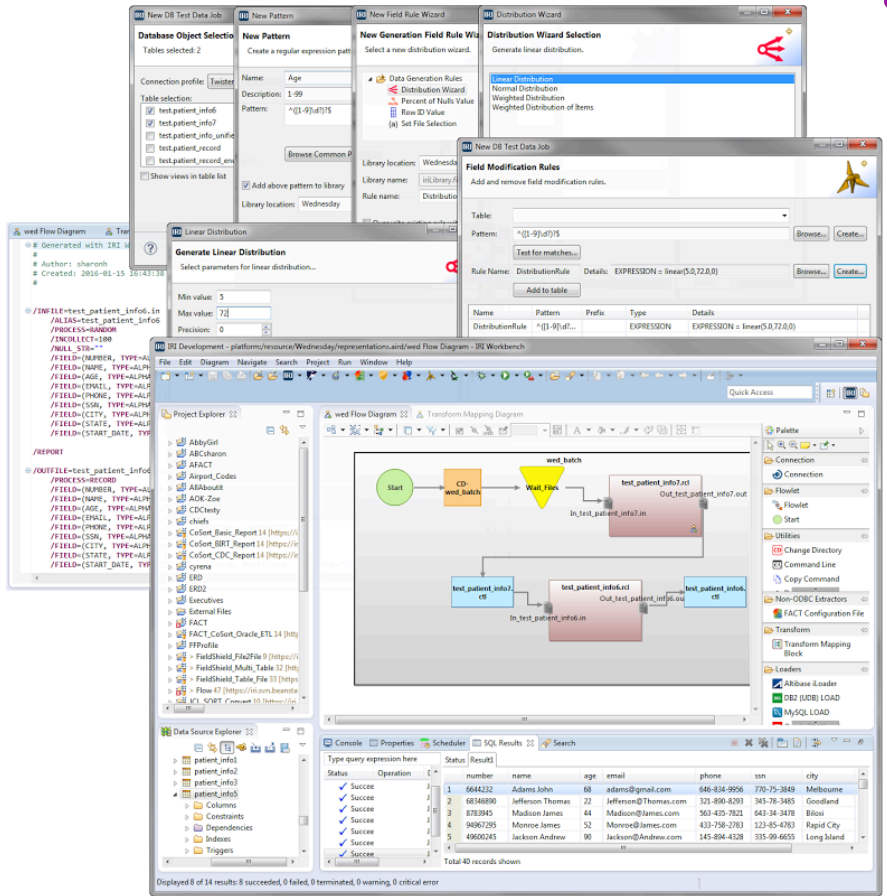
- Encryption and Decryption Functions:** A window showing a list of encryption functions like `enc_aes256(S(FIELDNAME))` and a configuration area for a new rule with fields for source name, passphrase, and function name.
- New Protection Field Rule Wizard:** A wizard window with a tree view of protection rules including Assignment Expression Builder, De-Identify or Re-Identify Function, Encoding or Decoding Functions, Encryption or Decryption Functions, Hashing Functions, Masking Function, Pseudonym Replacement, Random Value Generation, and String Manipulation Functions.
- Target Field Layout:** A window for mapping and modifying target data layouts, showing a table with columns for Field Name, Data Type, Position, and Size.
- Masking Function:** A dialog box for selecting a mask (e.g., Whole Field, Without Field, Credit Card) and defining pseudonym replacement rules with various options like name type, sex, and order.

# Test Data Synthesis & Population

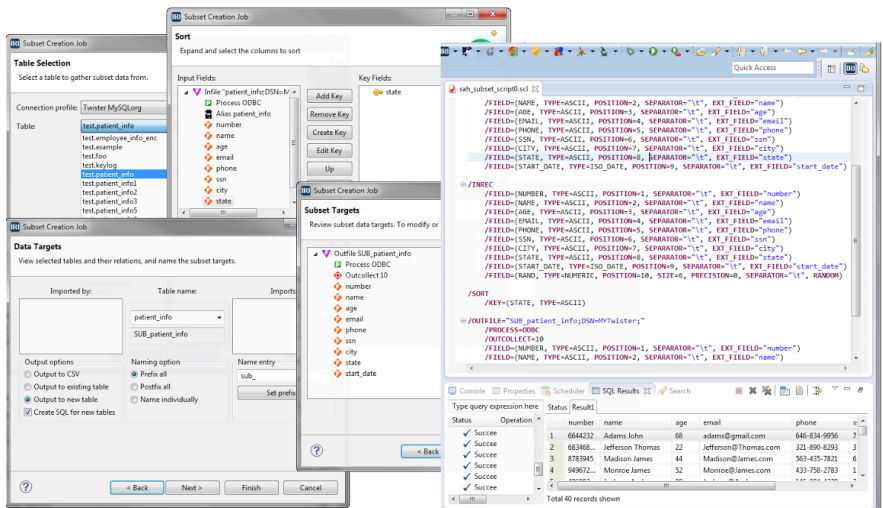


Voracity includes all [IRI RowGen](#) test data facilities for synthesizing and loading structurally and referentially correct [databases](#), plus [multiple file](#), Hadoop, SaaS, [Data Vault](#), and customized report targets. [‘Set file’](#) and job creation wizards help automate and modify the processes of parsing, generating, and delivering the test data to multiple consumers, or even virtualizing it for immediate [DevOps](#).

Through RowGen tooling in Voracity, you can randomly generate field values in more than 100 data types, or randomly select data from set files to combine artificial and randomized production data. [Test data realism](#) is also achieved with: composite form creation; null, range and frequency distributions, embedded data transformations; and, (report) [formatting](#) controls. There



# Database Subsetting



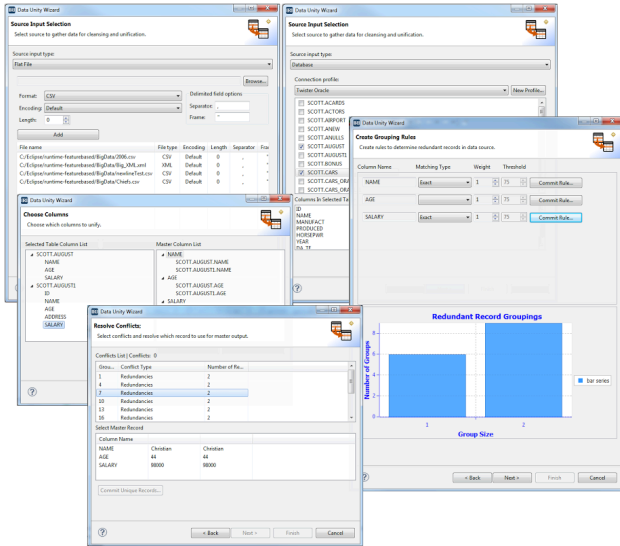
In addition to protecting production data and generating synthetic test data, Voracity users can also produce database test sets by creating plaintext or masked *subsets* of databases that maintain referential integrity.

The [DB subsetting](#) wizard in Voracity allows you to define the size, masking, and mapping of smaller copies of data for testing.

## Master Data Management (MDM)



Voracity supports the identification, matching, standardization, and protection of master customer and product data across disparate sources. Its MDM capabilities are domain-agnostic, and thus can also extend to master data for service, asset, supplier, financial, geographic, and other attributes across industries. Master data values and formats can be defined in user scripts that specify composite data types and format masks, or derived from existing sources. GUI wizards are also provided to unify data:



Voracity's consolidation-style MDM wizard scans, locates, analyzes and corrects redundant records in files and databases. To identify potentially redundant values, users can choose from Dice, Exact, Levenshtein, and Metaphone fuzzy matching algorithms. Once the conflicts in connected records are found, the wizard helps you rectify and save them.

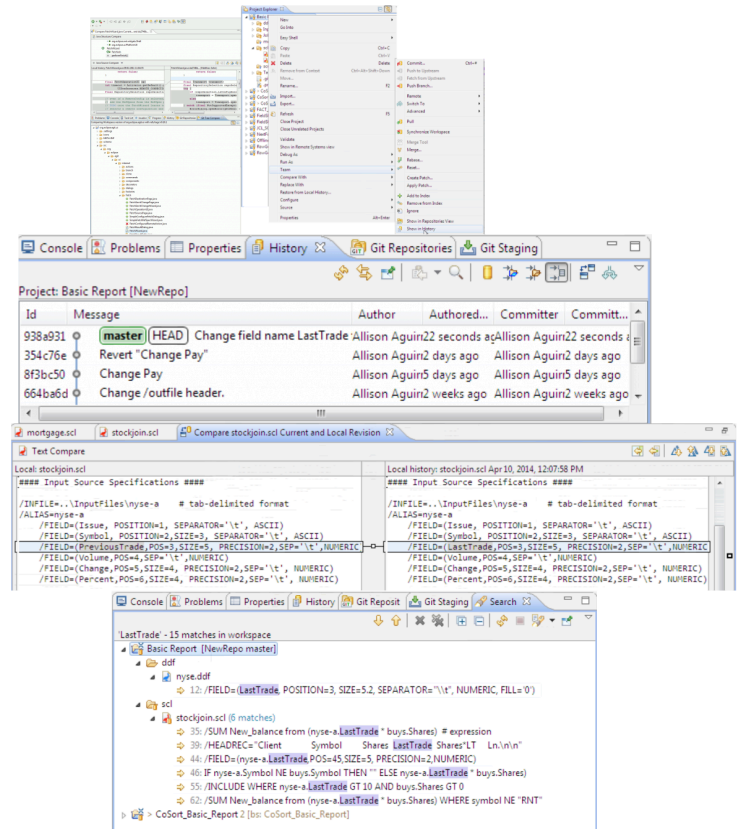
A planned registry-style MDM wizard also centralizes conflict value resolution, and then facilitates the operational propagation of the master data back into its original sources. Meanwhile, more advanced, custom MDM solutions built atop Voracity are possible. Consider [this inter-agency MDM use case](#).

## Enterprise Metadata Management (EMM)

Data and metadata undergo frequent change. As data gets modified in the course of Voracity activities like cleansing, unification, and remapping, so too can its metadata. The re-entrant metadata framework of Voracity simplifies [metadata management](#) and supports free [data lineage](#) inside IRI Workbench, or graphical output in erwin EDGE.

Voracity leverages the same, simple DDF for data layouts, and the same manipulation syntax that all IRI software uses; the open, explicit [metadata](#) of the core Voracity processing ([SortCL](#)) program.

Enforced metadata centralization supports the separation of data from applications, and the management of metadata assets in local or cloud [repositories like Git](#). Easily share, secure, version-control, and change-track DDF files, IRI job scripts, ETL workflows, launch configurations, SQL procedures, and other Eclipse project assets.



# Analytics





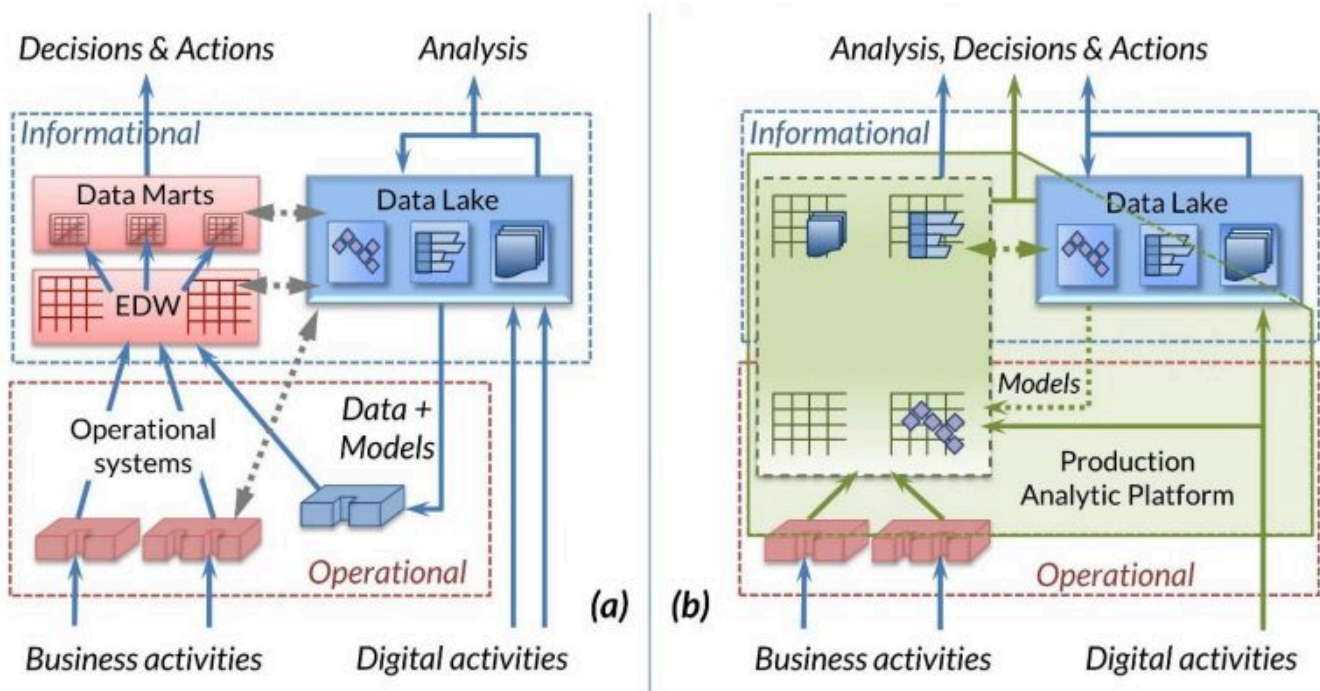
# Analytics

The final result of most data integration efforts is analytics, because credible visualization of insight is only possible after data has been properly and centrally prepared. Voracity combines data preparation, prediction, *and* presentation capabilities, which you can use in different ways; e.g.

1. pre-process data and send (franchise) it to another visual BI or analytic tool
2. process and present BI in the same 4GL and I/O pass using CoSort SortCL program
3. process and predict in SortCL and present BIRT graphs or charts at reporting time
4. self-service, big data BI and data science nodes in analytic and AI platforms like KNIME.

And because Voracity supports the *simultaneous* preparation and presentation of big data, it was dubbed a [Production Analytic Platform](#) by DW/BI industry founder Dr. Barry Devlin in 2018:

*“Over the past two decades, business people have increasingly demanded access to information, irrespective of its provenance, location, or format. Such demands raise real issues about the quality of the data sources and the ability of decision makers to understand the real meaning of the information used and limitations to its use. While IT generally understands—and often fears—these issues, we have nonetheless striven to facilitate unfettered access by business people to all available data. The Production Analytic Platform, shown in Figure 1b below, is a further response to these needs:*



*Data virtualization and batch ETL are at opposite ends of the timeliness spectrum of data preparation and integration. Bringing these processes together has been an elusive goal. Recognizing this reality, IRI offers data virtualization as a no-added-cost attribute of its Voracity ETL platform. The same language (SortCL) that is used to define its offline data preparation (filter, integrate, cleanse, mask, etc.) operations is also used to define real-time data virtualization.*

With Voracity, the metadata generated for the offline scenario is the same used in real-time virtualization, and vice versa. This is a mandatory data quality/integrity requirement when the same combination of data is needed in both ETL (for a stored copy) and via data virtualization (for an up-to-the-second view). This requirement is very difficult to achieve when separate ETL and data virtualization tools are used, because semantic differences between them can lead to confusion around business meaning and inconsistent virtualized and offline results.

In addition to ETL, Voracity offers virtualized data to business users through several approaches:

## Data Wrangling for Popular Analytic Platforms

By separating data integration from display, Voracity speeds time-to-insight for these tools:

<a href="#">BOBJ</a>	<a href="#">Cognos</a>	<a href="#">DWDigest</a>	<a href="#">KNIME</a>
<a href="#">Microstrategy</a>	<a href="#">Oracle DV</a>	<a href="#">Power BI</a>	<a href="#">QlikView</a>
<a href="#">R</a>	<a href="#">Splunk</a>	<a href="#">SpotFire</a>	<a href="#">Tableau</a>

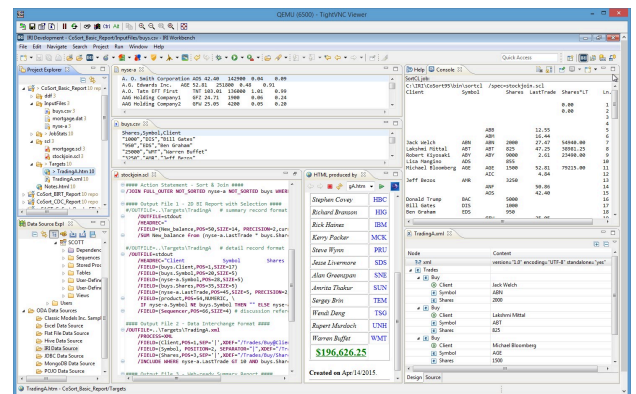
Voracity's fast extraction, transformation and loading capabilities 'franchise' data in the file system or Hadoop, and produce CSV, XML or ODBC subsets that these tools can rapidly ingest. [Benchmarks](#) show dynamic displays and what-if analysis can happen between 2 and 16x faster.

Centralized data preparation not only removes the burden of connecting to data, filtering and cleansing, sorting and joining, aggregating and calculating -- and sometimes masking it -- it also eliminates the redundancy and data synchronization issues of churning data every time a report or analysis is needed.

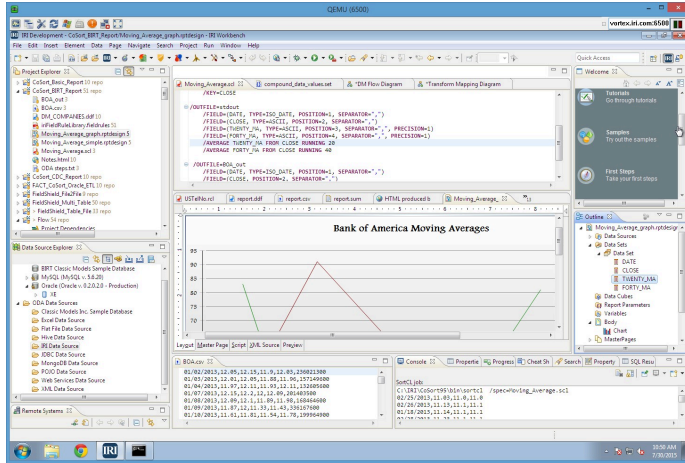
## Embedded BI & Reporting Features

Against any number and volume of [sources](#), Voracity can filter, transform, calculate, mask, and present data in one or more custom detail and summary reports in the same job script and I/O pass. Reporting [logic](#) and [layout](#) specifications are built in a text editor, or through wizards and dialogs in the Eclipse GUI for all IRI software, [IRI Workbench](#).

You can run these **descriptive** reports from: the command line, batch scripts, the GUI, other applications (via system or API call), or a job scheduler inside (or outside) of IRI Workbench.



# Embedded Analytics & Eclipse Visualizations



Voracity can prepare and pass data in memory through the Open Data Access (ODA) API for Eclipse for consumption by the Business Intelligence Reporting Tool (BIRT) or through a fit-for-purpose data source (Voracity provider) node for the KNIME Analytics Platform. Either way, Voracity wrangles static or streaming data when BIRT or KNIME run. This means that BI architects and data scientists can blend, govern, and display data *at the same time*, using the power of Voracity and open source analytics in the same pane of glass, IRI Workbench.

These Voracity-compatible integrations in Eclipse also deliver the ability to:

- profile multiple data sources to discover salient data and relationships
- collaborate with ETL architects on data metadata and preparation in the same IDE
- filter, transform, and run math, statistical, and trig functions -- all in the file system, not database
- get cleaner, richer, higher quality data -- to improve the analytic reliability
- federate (virtualize) data -- for ad hoc BI and what-if analyses
- mask sensitive data during transformation -- to safeguard PII in the reports

For fast **predictive** analytics, run a

1. Voracity CoSort job (script) that connects to, transforms, and applies embedded analytic functions (like linear regression, and Boost © confidence intervals) to your data sources ...

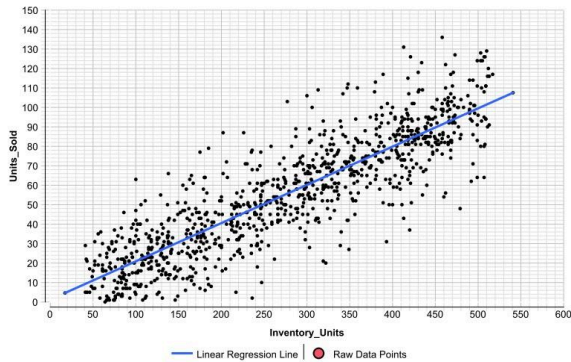
*simultaneously with:*

2. BIRT, to display a customizable plot and report with CoSort results sent through memory ...

to ascertain the trend line. Cluster and decision-tree for analysis, and eventually ensemble modeling, are planned, along with fit-for-purpose job wizards front-ending those jobs in the GUI.

For fast **prescriptive** analytics, design rule-based reports that **spot trends**; e.g., buy when the 20-day moving average crosses the 40-day average.

Unit\_Inventory\_vs\_Sales-2016-01-28\_09-11-30



LINEAR REGRESSION EQ:  $y = 0.19654 \cdot x + 1.23957$  R<sup>2</sup>: 0.75751

Source Data X			Source Data Y		
2-Sided Confidence Limits For Standard Deviation			2-Sided Confidence Limits For Standard Deviation		
Number of Observations	=	1000	Number of Observations	=	1000
Average X Value	=	276.8570000000	Average Y Value	=	55.6520000000
Standard Deviation	=	132.2689402354	Standard Deviation	=	257.7338250521
Confidence Value (%)	Lower Limit	Upper Limit	Confidence Value (%)	Lower Limit	Upper Limit
50.000	130.3408279853	134.3355967788	50.000	253.9767847034	261.7608271211
75.000	128.9787133938	135.7936444060	75.000	251.3226241484	264.6019188499
90.000	127.5862674885	137.3368770807	90.000	248.6132863588	267.6089986639
95.000	126.7152781512	138.3355024519	95.000	246.9121871872	269.5548790516
99.000	125.0377051831	140.3223342937	99.000	243.6433373945	273.4283379853
99.900	123.137198643	142.6878152838	99.900	239.9400518072	278.0366170993
99.990	121.5787005328	144.7251553885	99.990	236.9032629838	282.0054943599
99.999	120.2362521357	146.5527796333	99.999	234.2913202854	285.5667279083

Jan 28, 2016, 9:11 AM

For more advanced analytics, machine learning, deep learning (artificial intelligence), data mining, and **data science** projects, Voracity also feeds data directly to the **KNIME Analytics Platform** in Eclipse.

## Clickstream Analytics

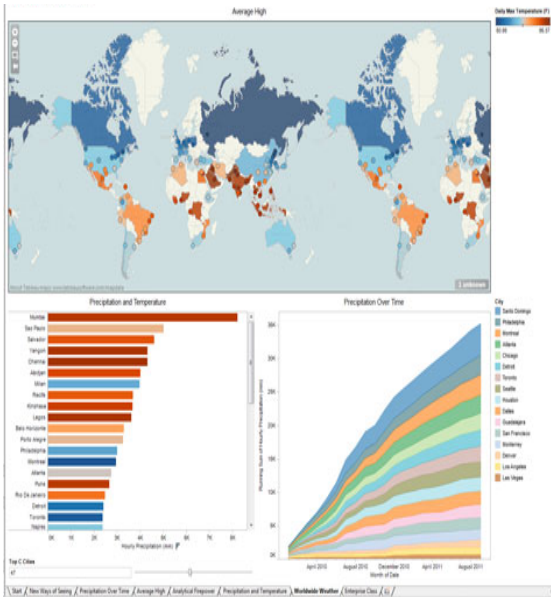


A clickstream data warehouse (CDW) or data webhouse supports retail and e-commerce decision making based on high-volume processing and analytics of [web logs](#). Voracity can:

- transform, reformat, and report on the log files
- apply selection logic to filter and segment data
- mask URLs, IPs, and other PII field values
- join related transaction data in other sources
- federate, convert, and multicast data subsets
- support and speed regular CDW refreshment
- populate BIRT or other visualization platforms.

Voracity's high performance data transformation algorithms and hardware optimization techniques optimize the efficiency of these operations in the file system or in Hadoop, without affecting online web or database operations.

Voracity-prepared data can also feed streaming sentiment analysis visualization tools in the cloud like JupiterOne.



## Customer Segmentation

Going against massive sources of customer and transaction data, Voracity simultaneously integrates and reports on [customer data](#) in distinct subsets, views, or filtered groups.

Powerful selection, deduplication, sort, join, aggregation, and re-formatting functions combine CDI, staging, and segmented reporting in the same job script and I/O pass. Create as many outputs as necessary, based on specified conditions, and in individually customized formats.

Apply different selection criteria for customer segmentation, including:

- name, age, address, or date ranges
- account or other identifying numerics
- transaction or product (SKU) IDs
- web page visits or IP addresses
- new, changed, deleted, or duplicate data

Uses built-in, field-level encryption (or other privacy protection functions) to prevent the exposure of sensitive data on a need-to-know basis. Each run produces a complete XML audit log so you can examine the user, job, and runtime parameters, providing detective control, and verifying compliance with privacy regulations.





# Voracity Curation (Data Lifecycle Management)

Data life cycle management (DLCM) is the process of curating data ... from planning its requirements all the way through its use and retirement. The cycle can manifest across applications, databases, and storage media. From profiling existing data to defining new data types, data mapping and masking to analytics, and from metadata management to master data management, every stakeholder can share Voracity's GUI and exploit data as a team.

## **Profile & Acquire**

Discover and extract data and metadata, and auto-create IRI-ready metadata, as you connect to one or more of data sources. Define custom data structures, mask date formats, test values and distributions.

## **Cleanse & Unify**

Filter, enrich, scrub and standardize data in multiple sources. Select, fuzzy-search, and merge reference data into master tables and values.

## **Protect & Audit**

De-identify data at the field level as you filter, transform, report on it, or hand it off. Use FieldShield functions to encode, encrypt, hash, pseudonymize, redact, tokenize, etc. Produce an automatic machine-readable audit log from any job for query-ready reports or SIEM display of the runtime environment and job parameters. Score the risk of re-identification from masked data or unmasked quasi-identifiers to comply with the HIPAA Expert Determination Rule or FERPA and use the results to further anonymize data to render it compliant but still useful for research and marketing purposes.

## **Process & Provide**

Integrate, migrate, govern, and analyze data in the same job and I/O pass you can design and deploy in the same 'pane of glass' -- the Voracity graphical IDE, IRI Workbench, built on Eclipse. Designate and feed multiple test or production targets in table, file, report, cube, DB load and/or DataVault format.

## **Express & Predict**

Use Voracity-embedded BI features to cull and group, and apply math, stat, and formatting, to turn raw data into detail, summary, and trend reports, or hand-off wrangled results in files or tables for your BI tool. Or, use included hooks to free BIRT or KNIME in Eclipse to seamlessly stage, analyze and display.

## **Convert & Replicate**

Rapidly reformat legacy data sources and data types, or simply specify new target formats in any operation. Create copies and subsets of data in any number of formats at once with simple GUI dialogs, or use Voracity's myriad ETL design options for more elaborate data replication.

## **Publish & Share**

Beyond your options for directing production results or test data to persistent or federated (virtualized) targets, you can connect others to them in shareable repositories like Git, etc. Version-control your data and metadata, track their changes and lineage, and secure access to both in the cloud.

# Technical Specifications

Voracity is a general-purpose data management suite for big data discovery, integration, migration, governance, and analytics. Its native data movement, manipulation and metadata utilities, API libraries, and hooks to third-party software, all leverage the common metadata of the CoSort 'Sort Control Language ([SortCL](#))' program, and are supported in a graphical IDE built on Eclipse™ ([IRI Workbench](#)).

Voracity includes the features listed in [this matrix](#), and most of the functions of the IRI product line exposed in the Workbench GUI. Following is a non-exhaustive list of included platform functionality:

## Installation

- Distributed via internet or user-specified media
- Menu-driven setup and configuration utility for the CLI
- Separate steps for FACT, Hadoop, Erwin (AnalytiX DS), and external (3rd-party) components

## Invocation

- Command line (including pipe sequences), shell commands, and batch scripts
- IRI Workbench (Eclipse GUI) 'Run As' menu, 'Run Configuration', or built-in [task scheduler](#)
- API calls to the CoSort SortCL executable or `sortcl_routine()` library
- Third-party automation (scheduling tools) like cron or [Stonebranch Universal Controller](#)

## Data Discovery (Profiling)

- Searches for user-specified literals (string values), strings in a dictionary look-up file, or values conforming to patterns expressed in regular expression syntax across flat-files, JDBC-connected data sources or entire schemas, and "dark data" document file formats
- Extracts values and metadata from document file formats (HTML, MS Office, PDF, RTF, TXT, XML) based on regular expressions into a flat file and creates an IRI DDF file for it
- Generates statistical reports on selected attributes of flat-file and JDBC data sources
- Validates relationships (checks for referential integrity) between JDBC-connected tables
- Produces E-R diagrams for relational databases connected through JDBC
- Auto-creates or supports user-defined field definitions in IRI DDF syntax for flat files

## Input and Output

- Processes any number of files and ODBC sources listed [here](#), of any size, and any number of records, fixed or variable length (to 65,535 bytes) passed from those sources, as well as input procedures, stdin or named pipe, cloud storage, HDFS, S/FTP, HTTP/S URLs, MongoDB collections, MQTT/Kafka, or applications. Contact IRI regarding REST and web service calls.
- Supports environment variables and wildcards in source and target specifications, along with absolute and relative path names, ODBC DSN files, aliases, and URLs
- Accepts and outputs fixed- or variable-length records with delimited fields
- Creates detail, delta, and summary report targets, or hand-off files or tables for BI tools
- Outputs sequence (index) numbers in each row, at user-defined start and interval values
- Builds set files from DB columns or user-defined, composite formats and value ranges
- Randomly generates, and/or randomly selects (from set files), test data values
- Returns processed (transformed, masked, formatted, etc.) records one (or more) at a time to a file, pipe, output procedure, formatted report, database table, and/or application

## Record Selection and Grouping

- Includes or omits source and target data by comparing fields to each other or constants, or through SQL /QUERY (SELECT) logic
- Compares on any number of fields, using standard and alternate collating sequences
- Sorts and/or reformats groups of selected records
- Matches two or more sorted or unsorted sources on inner and outer join criteria
- Skips a specified number of records, bytes, or a record header
- Processes a specified number of records or bytes, including a saved header
- Eliminates or saves records with duplicate keys.

## Sort Key Processing

- Allows any number of key fields to be specified in ascending or descending order
- Supports any number of fields from 10 to 65,535 bytes in length
- Orders fields in fixed position or floating (on one or more delimiters)
- Supports numeric keys, including all C, FORTRAN, and COBOL data types
- Supports single and multi-byte character keys, including ASCII, EBCDIC, ASCII in EBCDIC sequence, Thai characters, and natural (locale-dependent) values
- Supports American, European, ISO and Japanese timestamps
- Supports Unicode and double-byte characters like Big5, EUC-TW, UTF-8 and 16, and SJIS
- Allows left or right alignment and case shifting of character keys
- Accepts user compare procedures for multi-byte, encrypted and other special data
- Performs record sequence checking
- Maintains input record order (stability) on duplicate keys
- Controls treatment of null fields when specifying floating (character separated) keys
- Collates (and converts between many of) the data types (formats) described [here](#).

## Record Reformatting

- Inserts, removes, resizes, and reorders fields, or columns within rows where permitted
- Derives new field values through the use of various field-level functions
- Converts data in fields from one format to another either using internal conversion
- Maps common fields from differently formatted input files to a uniform sort record
- Performs mathematical operations and functions on field data (including aggregated data) to generate new output field values
- Joins any fields from several files into an output record, usually based on a condition
- Changes record layouts from one file type to another, including: Line Sequential, Record Sequential, Variable Sequential, Blocked, Microsoft Comma Separated Values (CSV), ACUCOBOL Vision, Micro Focus I-SAM, MFVL, Unisys VBF, VSAM, W3C Extended Log Format, LDIF, and XML
- Supports Boost-compatible date masking and composite data templates to build new data types
- Maps processed records to many differently formatted output files
- Writes multiple record formats to the same file for complex report requirements
- Resolves data quality conflicts and unifies values in files and databases through consolidation and registry-style master data management (MDM) wizards using Dice, exact, Levenshtein and Metaphone matching algorithms with user-defined probability weightings.

## Data Cleansing, Reformatting, Protection and Validation (Governance)

- Retrieves and re-maps values from multidimensional, tab-delimited lookup files on the basis of equal or conditional matches (suitable for Slowly Changing Dimensions)
- Creates and processes sub-strings of original field contents, where you can specify a positive or negative offset (from the left or right, respectively, of the source field) and a number of bytes to be contained in the sub-string
- Finds a user-specified text string in a given field, and replaces all occurrences of it with a different user-specified text string on output
- Analyzes fields to display the offset number for the specified occurrence of a string
- Manipulates and displays literal values with input data inside field statements for use in value derivations, functions, conditions, cross calculations, and reporting
- Aligns desired field contents to the left or right of an inrec or output field, where leading or trailing fill characters from the source are moved to the opposite side of the string
- Supports Perl Compatible Regular Expressions (PCRE), including pattern matching
- Uses C-style “iscompare” functions to validate field contents (e.g., for printability) for use in /INCLUDE and /OMIT record filtering statements
- Protects sensitive data with field-level de-ID, deletion, and multiple encryption routines, as well as anonymization through blurring or bucketing, pseudonymization, randomization, hashing, filtering, string manipulation, omission, character obfuscation, and other data masking methods
- Encryption functions can be either user-specified functions or Voracity-built-in libraries: 3DES, AES-128 and 256 bit (including format preserving), GPG, and FIPS-compliant OpenSSL
- Supports custom, user-written field-level transformation libraries, and documents examples of advanced cleansing (e.g., address standardization) routines from Melissa Data and Trillium.

## Record Summarization

- Consolidates records with equal keys into unique records, while totaling, averaging, or counting values in specified fields, including derived (cross-calculated) fields
- Produces maximum, minimum, average, sum, and count fields
- Breaks on single or compound conditions
- Displays running summary value(s) up to a break (accumulating aggregates)
- Ranks data through a running count of descending numeric values
- Calculates moving averages, multiplication, standard deviation, and linear regression
- Allows multiple levels of summary fields in the same report
- Write detail and summary records to the same output file for structured reports
- Remaps summary fields into new formats and data types for different relational targets.

## Information Discovery (Analytics)

- Produces detail and summary reports
- Produces change capture reports without DB logs
- Produces slowly changing dimension reports
- Produces trend-line and intersection reports
- Feeds KNIME nodes in Eclipse through a free Voracity data/job source provider
- Produces data for in-memory, ODA driver consumption by BIRT at reporting time
- Creates CSV, JSON, XML, or ODBC hand-offs for third-party BI and analytic tools.

## Metadata Controls

- Generates field, or data definition files (DDF) automatically from database tables, delimited files, the metadata discovery wizard, and new job wizards
- Converts native COBOL copybook, Oracle SQL\*Loader control file, CSV, and W3C extended log format (ELF) file layouts into DDF
- Modifies DDFs in syntax-aware text editor, fit-for-purpose form editor, and multiple dialogs
- Supports the generation and modification of FACT .ini, and all SortCL-compatible IRI control language scripts (.scl, .ncl, .fcl, and .rcl) in syntax-aware text editors, form editors, and dialogs
- Supports the generation and modification of SQL select statement and procedures
- XML workflows, SortCL/RowGen .CL, and DDFs are supported by Erwin Mapping Manager, and DDFs are supported in the Meta Integration Model Bridge (MIMB) suite
- Saves all metadata assets in project folders for copying, sharing, printing, etc.
- Interacts with Git, SVN, CVS and other Eclipse-compatible version control systems.

## Resource Controls

- Sets and allows user modification of the maximum and minimum number of concurrent threads for sorting on multi-CPU and multi-core systems
- Uses a specified directory or a combination of directories for temporary work files
- Limits the amount of main and virtual memory used during sort operations
- Sets the size of the memory blocks used as physical I/O buffers
- Toggles the compression of work files to improve throughput
- Designates resource allocations at the system, user, or job level hierarchically
- Displays warning, error, and event data (at various levels of verbosity) to stderr at runtime
- Logs performance and application statistics, and produces an XML audit file, with each run
- Separate gateway tooling for Hadoop cluster configurations.

## Ease of Use

- Uses a self-documenting 4GL based on the VMS sort utility and SQL to define and process data
- Leverages centralized application and file layout definitions (metadata repositories)
- Incorporates all design and deployment in the familiar, graphical IDE of Eclipse
- Provides context-sensitive dialog help, and syntax-aware metadata (layout and script) editing
- Toggles runtime warnings
- Reports problems to standard error when invoked from a program, or to an error log
- Runs silently or with verbose messaging without user intervention
- Allows user control over the amount of informational output produced
- Generates a query-ready XML audit log for data forensics and privacy compliance
- Supplies first steps, demo projects, and tutorial 'cheat sheets' to facilitate onboarding
- Links to how-to blog articles and YouTube videos for advanced applications
- Describes commands and options through Eclipse help and .pdf documentation
- Easy-to-use interfaces and seamless third-party sort replacements preclude the need for training classes; however, advanced training is available in Florida or at user sites
- Phone, web, and email support available directly from the product developers
- Local language support is available from more than 40 international offices.

## Licensing Information

IRI and its worldwide representatives license and support Voracity on an affordable, [tiered subscription](#) basis. Technical support and software updates are included, while ad hoc implementation services are offered through IRI or authorized partners.

Refer to the Voracity product-feature [matrix](#) for 'pay for play' options. U.S. schools, non-profit institutions and government agencies qualify for additional discounts.

To encourage Voracity adoption and to support learning, both IRI Workbench and freemium editions (pending) are non-nodelocked. IRI also provides a free Java API called "GulfStream" for integrating external software with Voracity metadata. GulfStream is best exploited in (although does not require) Eclipse, because it also supports the creation and consumption of Workbench workflows.

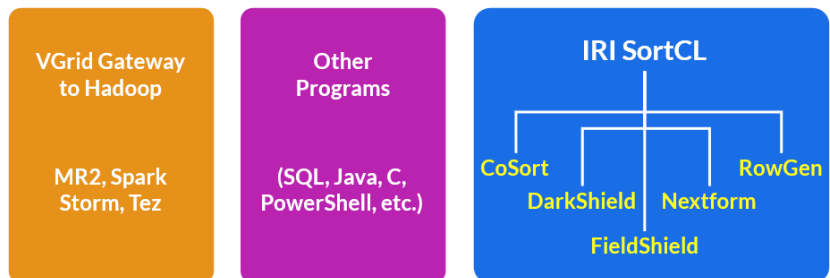


## Architecture

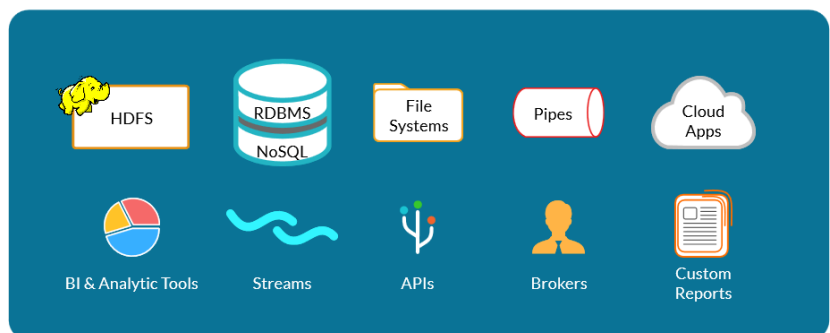
### Job Designers



### Runtime Engines



### Sources & Targets



## Professional Services

An IRI professional services engagement allows you to leverage more than 100 collective years of IT and integrated data-handling experience. Among the available services are:

- Big data preparation – packaging, protecting, and provisioning structured and unstructured data sets for analytic/BI, database (DB), and ETL operations
- Data masking – profiling, de-identification, encryption, tokenization, re-ID risk scoring, and other services to aid data loss prevention, data governance, and data privacy law compliance efforts
- Data replication and federation – acquiring, re-mapping, and creating virtualized views
- DB migration – mapping table data and relationships to new versions or platforms
- Data conversion – re-formatting JSON, LDIF, XML, and COBOL index files (e.g., Vision, I-SAM), multi-byte character sets, most mainframe data types, and endian states
- Master data management – value and format definition, quality validation, and security
- Program replacement – translating cryptic and inefficient SQL, 3GL, ETL, legacy sort, and shell procedures into IRI's simple, portable, 4GL text scripts
- Test data management – end-to-end services from DevOps needs definition through data generation and target persistence (without using production data)



# IRI Voracity

An Insatiable Appetite for Data

[LIVE DEMO](#)

[FREE TRIAL](#)

**INNOVATIVE ROUTINES INTERNATIONAL (IRI), INC.**

2194 Highway A1A  
Melbourne, FL 32937 USA  
Phone +1 321-777-8889  
<https://www.iri.com/voracity>

