seeCommerce

Data Integration Approaches For IT Managers

A White Paper by SeeCommerce
September 2001

3420 Hillview Ave., Lobby 8, Palo Alto, CA 94304
1.800.255.9520 • 1.650.213.1800

INTRODUCTION **3**

DATA INTEGRATION APPROACHES **4**

GENERAL PLANNING 6
MIDDLEWARE 7
POINT-TO-POINT EXTRACTION, TRANSFORMATION, AND LOAD 9
CAVEAT FOR MIDDLEWARE AND POINT-TO-POINT 10
FORMATTED TEXT FILES 10

SEECOMMERCE APPROACH **11**

DATA LOADING 12
FLEXIBLE CONNECTIONS 13
METADATA 13
JOB CONTROL 14
LOGGING 16
VERIFICATION AND TRANSFORMATION 17

SEECOMMERCE'S SUMMARY **18**

SCALABILITY 18
CONSISTENCY AND INTEGRITY 19
LOW LATENCY 19

FOR MORE INFORMATION **19**

This paper is intended for Chief Information Officers (CIOs) and members of Information Technology (IT) organizations. It explains common data integration approaches and the data integration framework used by SeeCommerce™.

Introduction

Presenting information so that timely and intelligent business decisions can be made is a critical IT role. That role involves effectively integrating disparate sources such as Enterprise Resource Planning (ERP) systems, Warehouse Management Systems (WMS), Advanced Planning and Scheduling systems (APS), Manufacturing Execution Systems (MES), Excel spreadsheets and other direct sources on a path into a target system.

This paper will discuss SeeCommerce's framework for acquiring raw data from source systems and moving it across network and system boundaries to enable sophisticated supply chain analysis on information from virtually any type (and combination) of organizational data. Scalability, Consistency, Integrity and Low Latency characterize this framework. The qualities that are desirable for achieving the best data integration process are:

Scalability – The need for the process to handle greater numbers of transactions and handle voluminous amounts of data brought about when the number of transactions, applications and users escalates.

Consistency – The repeatable execution of pre-defined mapping from source to target systems to achieve a common business model.

Integrity – The assurance that all data is complete, normalized, and reliably transported. Normalization is typically a refinement process after initially identifying the data objects that should be in the database, and their relationships, plus defining the tables required, and the columns within those tables. In general, the process entails the removal of redundant attributes, keys and relationships for accurate database results.

Low Latency – The reduction in the lag time from when data is created or changed until it is fully propagated to other systems and available for presentation defines latency. Whereas low latency is generally desirable, business requires

data at different time frequencies, so you may want small data sets as they are generated, and larger data sets handled at appropriate intervals to utilize off-peak hours. Ability to schedule can be an important factor in this area to make the data acquisition rate and frequency commensurate with the level of urgency.

By supporting multiple data integration technologies, SeeCommerce is able to serve many customers in a wide range of business environments with differing needs and infrastructure. This will be demonstrated by a discussion of general approaches to data integration followed by the SeeCommerce process for moving raw data through to business user presentation.

Data Integration Approaches

As illustrated in *Figure 1 From Source Through Data Aggregation*, various approaches can be used to solve the integration problem. There are three general techniques, which can be used alone or in combination:

Middleware - Route all communications through a common integration system. If an organization already has Enterprise Application Integration (EAI) or messaging infrastructure in place, this can be an elegant and highly effective integration solution. However, not all organizations have invested in middleware systems.

Point-to-Point - Connect individual source data systems to the analytic data store through an Extraction, Transformation and Load (ETL) tool. This technique is easy to understand and is often the easiest to implement. In situations where it's appropriate, the only required infrastructure is that the source system be network-accessible from the analytic data store.

Formatted Text Files – These are files containing one record per line with fields on each line delimited by tabs, commas or some other reserved character. Appropriately used, this common technique can be reliable, pragmatic, and effective.

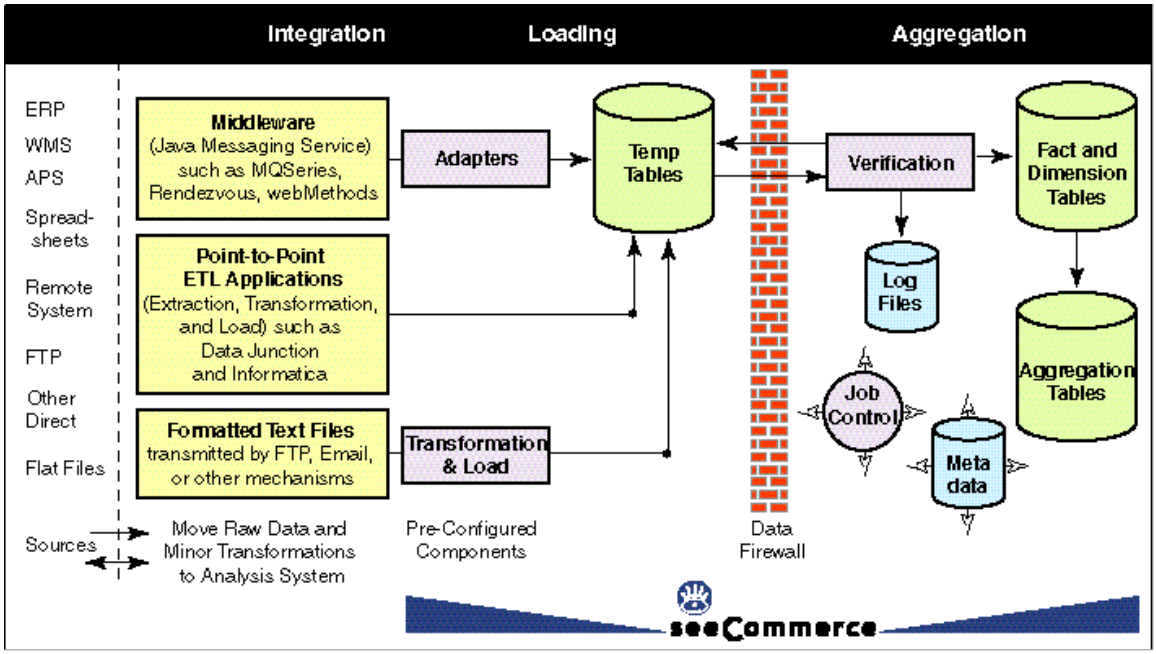

Figure 1 From Source Through Data Aggregation

Any technique for data movement obviously has two essential steps: Acquiring the raw data from the source system in one format, and delivering it to the destination system in another format.

Because of the variety of ways that data can be transferred into the analytic data store, the foundation must start with the definition and implementation of the data interface framework in the analytic data store itself. That framework should include these characteristics:

- Data loading capabilities to synchronize data coming from different sources at different times, and to hold it for processing until a logically complete set is received.
- Flexible connection strategy, so that data can be exchanged from a variety of different sources and formats.
- Metadata that captures the content, sources, dependencies and other rules describing how a particular set of data fits into the framework.
- Job control mechanisms so that the various phases of the data integration process can be coordinated or executed as needed.
- Logging mechanisms so that intermediate and final results can be determined both automatically and by an operator.

- Verification and transformation mechanisms to verify data integrity, enforce dependency rules, and transform the content as needed.

GENERAL PLANNING

There is a certain amount of generic planning required for data integration, regardless of the sources and destination tables involved. The following list is a guide applicable for any circumstance:

- Decide on the type and format of data that the analysis tool needs to receive.
- Resolve the update rate needed for each data subset. For instance, some data might be low volume and very stable—it might be sufficient to transfer that information weekly. Other data might be high volume and dynamic—it might be desirable to transfer incremental updates daily, hourly, or faster. For event-driven data, estimate the average and peak frequency at which the events might occur.
- Estimate the volume of data of each type that will be moved during each update cycle.
- Determine the source systems that contain the original data.
- Investigate any interactions between original data sources. For instance, to analyze fulfillment metrics the analytic data store might need the order date, quantity, ship date, and quantity relating to a particular customer transaction. Depending on existing data systems, it's possible that the information is in two or more separate source systems.
- Capture all of this information as metadata to insure repeatability of this process.

For each source system and data set, you can then determine which of the three previously mentioned approaches (middleware, point-to-point, or formatted text files) is the most appropriate. It's possible that different strategies will be used for different data sets—even within the same organization.

MIDDLEWARE

Organizations with middleware installed will often prefer to move everything through that system. There can be a number of advantages to middleware. The content is reliable due to its continuous connection from source to destination. Since it is often event-based, it allows for "as fast as possible" response without continuous polling of the source system. It moves data easily across physical boundaries of distance. With the proper infrastructure, the source and destination systems can be thousands of miles apart, but they can be as well connected as if they were in neighboring offices. Depending on the system and usage, it may support transactions that group related operations.

Middleware almost always reduces the total number of interfaces in a system. Each system has just one connection—to the messaging system—rather than various systems connecting individually to each other. The only data that can be read from or written to the source system is that which is specifically configured in the middleware system, thus preserving source system isolation and security. Once a given set of data is flowing into the system, it can be accessed by any number of other consumers for that same data. The next sections describe ways in which middleware can handle the data acquisition, conversion, and delivery as well as the steps taken in planning a middleware implementation.

Data Acquisition

Depending on the system and how it's configured, middleware systems can either read data directly from the source system, or accept data that is written to them from the source system. If data is written to the messaging system, it often will be scheduled by the source system itself, or run when events occur. If data is read by the messaging system, it is often because of timed configuration in the messaging system, or because of a request from the destination system.

Data Conversion

Once the data is received by the messaging system, the behavior again depends on the system capabilities and configuration. One common approach is to deliver the data immediately to one or more subscribed destination systems, if

they're running. If the receiver isn't running, the data might be discarded.

Another approach is to store the message into a database within the messaging system itself. It can then be held for any amount of time, waiting for the receiver to pick up the data. This general approach is often set up with recoverable, guaranteed delivery. In this case, the messaging system guarantees that once the message has been sent, it will eventually be received, even if the receiver isn't currently available, or if any of the systems (source, middleware, or destination) crashes or is taken off-line. Although obviously more robust, the storage and performance implications of recoverable, guaranteed-delivery systems make tradeoffs necessary.

Data Delivery

Again, behavior is flexible depending on the type of system and how it's configured. Middleware systems can deliver data directly to the destination—for instance writing rows of data to a database. Or they can hold the data until requested to move the data by the destination system. Direct storage is usually more efficient for large sets of data.

Middleware Planning

Here are some planning tips for using this method:

- Refer to data interface documentation that is produced by SeeCommerce.
- Determine what needed data may already be flowing into the middleware system.
- Determine the sources for additional data not already flowing into the system and add the additional sources as needed.
- Identify or define the taxonomy associated with the data. Depending on the underlying system, this might be the subject to which that data is published, or the message queue name to which the data is sent, as well as the format of the transmission itself.

POINT-TO-POINT EXTRACTION, TRANSFORMATION, AND LOAD

Organizations that don't have middleware infrastructure in place but have fairly conventional source systems that are supported in most ETL tools, like Informatica, will often choose to use an ETL tool for data integration. Since ETL's task is simply to pick up data directly from the source and deposit it in the destination system, it is easy to understand. The output can be directly into the destination database tables from source-to-destination in a single step. ETL works with a variety of data sources and can apply various transformations. It requires minimal existing infrastructure, just a network connection between source and destination. A further consideration is that many applications have APIs for programmatic execution initiated by the analytic data store. Data integration using ETL tools is profiled next.

Data Acquisition

The ETL system is executed either via a program or by an operator. It then reads the defined set of data from the source system.

Data Conversion

ETL systems generally have some kind of transformation language defined. However, they generally have no storage mechanism and require source and destination systems to have a direct electronic connection working at the time of the conversion.

Data Delivery

The ETL system writes the data directly to the destination system. If it's invoked by a program, then the calling program will normally have some mechanism for determining whether the conversion was successful or not, and dealing appropriately with the result.

Point-to-Point (ETL) Planning

Here are some planning tips for this method:

- Refer to data interface documentation that is produced by SeeCommerce.

- Determine the sources for each data set.
- Use the ETL's design tools to define the source mapping for each data set, and any necessary minor transformations.
- Use the ETL's design tools to define the destination mapping for each dataset. Note that source data sets do not necessarily have direct mirrors to the destination system.

CAVEAT FOR MIDDLEWARE AND POINT-TO-POINT

Middleware and ETL tools of various kinds often have a simple GUI for configuring data selections. Although they work fairly well, sometimes the result isn't scalable. The reason is that source systems are frequently quite large, and the incremental data needed in a particular update cycle is often quite small. For instance, if in an hourly update the application needs only the most recent 1-hour of data, but there are 10 years of data in the source system, it's obviously impractical for the ETL tool to select the entire data set and apply its own selection criteria. Instead, the integration must be planned using the source system's native selection syntax and making use of appropriate indices so that the needed data can be acquired at a reasonable performance cost.

FORMATTED TEXT FILES

Formatted text files are among the simplest of techniques. Formatted text files are a workable solution that can be applied in virtually any situation. Depending on the assignment, it might be a preferred approach, or it might be a choice of last resort, but either way it can be made to work effectively. This is the obvious choice for disconnected systems—those that do not have a direct electronic link to the source system, but can generate a text file and send it by FTP or e-mail, for instance. A data acquisition, conversion and delivery overview for formatted text files is the last in this series of general approaches for data integration.

Data Acquisition

The text files are created by some kind of utility program to read data from the source and create the text file. There are a number of ways this is accomplished, such as utilizing ETL tools, the source system, or a manual process to produce appropriate output files.

Data Conversion
Once in file form, there are obviously no additional transformations applied to the files, but the files themselves can be stored as needed. If named and archived appropriately, they can be as persistent as need be. Movement of the files can be accomplished by any technique that can move files, such as FTP, e-mail, network file copy, backup/restore and others.

Data Delivery
Once the file is within the network visibility of the destination system, the destination system itself must have appropriate software to read the files and deposit the content into its own database.

Formatted Text File Planning
Here are some planning tips for this method.

- Refer to the data interface documentation that is produced by SeeCommerce.
- Determine the sources for each data set.
- Assess if you already have existing flat files between operational systems. If so, SeeCommerce can map easily to their existing format. If not, use any convenient approach for creating the necessary files. In many cases, an ETL tool can be used, even though the output is a formatted text file instead of direct input into the destination system.
- Plan how to transfer the files. SeeCommerce has some mechanisms such as FTP built in, but any convenient method can be used.

SeeCommerce Approach

Our implementation includes a core framework for reliable data integration, regardless of the type of source systems or infrastructure framework that our customers might have. In addition to the core framework, we have various adaptors and

utilities that optimize the use of selected middleware, point-to-point and formatted text files.

The SeeCommerce data integration framework is completely consistent with the elements identified previously (data loading, flexible connections, metadata, job control, logging and verification) for a robust strategy. We bring a number of positive attributes to the process in these areas and they are the subjects of this section.

DATA LOADING

SeeCommerce implements loading features to synchronize data that may be received from various data sources, using different integration techniques, at different rates and times. One key feature of the implementation is a temporary data area with a dedicated login and table space. This acts as a holding area for data where external processes such as ETL tools and messaging systems can deposit data when ready. Data is held in the loading area until dependent data can be transferred (coordinated using features in See Commerce's Job Control module) and the data set is ready to be processed by the SeeChain® applications.

Since this is a separate login and table space from the destination database, the loading area also acts as a data firewall. Data is not allowed to cross the firewall until the SeeChain™ Application, running on the other side of the firewall runs verifications and pulls the data through.

In the SeeCommerce data integration framework, ETL and middleware tools can store data directly into the temporary tables. For selected middleware tools, we provide connectors for attaching to the middleware system and storing message content to the temporary tables. We also have utilities for parsing formatted text files from several common formats to store the results in the loading area. See *Figure 1 From Source Through Data Aggregation*.

FLEXIBLE CONNECTIONS

Assisted by the simple, open-definition data format, many data transfer tools can transfer data directly into the temporary table area, including ETL tools, EAI tools, and many messaging tools. Any tool that can read and write an Oracle or DB2 database can exchange data directly with the SeeChain database. In addition to direct table storage, other connection techniques are supported:

- **Formatted text files** - We provide a utility for parsing and verifying files in common delimited formats and storing the results into the destination temporary tables.
- **Messaging** - We provide a Java Message Service (JMS) client which allows us to connect to any messaging system that defines a JMS interface, and receive events or data as JMS messages. We support both publish/subscribe and request/response models for moving the data.
- **webMethods** - We provide a set of pre-configured webMethods "Intelligent Adaptors" that can be used either as-is, or modified for specific circumstances.
- **ETL tools** – We provide Java-based integration with selected ETL tools to utilize existing ETL tool investments for extracting data from operational systems.
- **Command Line** – Many ETL or other tools can be executed from the operating system command prompt. We provide a mechanism for defining a connection in terms of a command line, allowing integration with any such tool.

METADATA

Data for SeeCommerce's SeeChain applications are organized as a dimensional data store. As such it contains "fact" data (information about specific transactions occurring in your business operations) and "dimension" data (information about those transactions). Facts and dimensions often come from different operational source systems, and at different times. The SeeCommerce implementation handles this complexity by capturing metadata defining each set of fact and dimension data, including:

- The exact content expected for any one set of data—defining each of the fields, the type of data

expected in each field, whether the field is required or optional, etc.
- The dependencies between different sets of data, for example, the "dimension" data sets that a particular "fact" set depends on.
- Source and destination for each data set.
- Scheduling information defining when each data set is to be acquired, for instance, on a timed-basis, event-basis, or asynchronous reception from an outside source.

The metadata describing the different data sets are defined and stored in the SeeCommerce metadata repository as part of the overall SeeCommerce installation and implementation process. In addition to the internal configuration of the SeeCommerce data store, we also generate metadata that defines the data transfer process:

- **Formatted Text File Transfers** - Tables are generated that define the mapping between the text files and the temporary tables.
- **JMS Connections** - Configuration tables are generated that define the expected layout and handling of the messages that might be received. The content of messages can be defined either as named fields in the message itself, or the message content can be parsed as a delimited or XML data buffer.
- **webMethods Integration** - We provide a customized webMethods adaptor that reads the metadata for more convenient configuration using the webMethods' mapping tools. We also provide sample configured adapters for many typical source systems.
- **Documentation** - Documentation for administrators and users is generated to facilitate any other operations that need to be configured in external systems.

JOB CONTROL

Job control works in conjunction with data sets, coordinating data transfers between internal and external tools. As a generalization, a "job" is usually one data transfer operation, and the states and history of that transfer definition. Some of the key states include: **allowed** (an external or internal tool is allowed to perform this data transfer), **in progress** (a tool is

executing this transfer right now), and **complete** (a previously executing transfer has finished).

Although external tools such as ETL tools or other middleware might have their own job control features, the significance of these features in the SeeCommerce system is that it is a facility for coordinating operations. These can be between different tools, between operations that are internal to SeeCommerce and external to SeeCommerce, or that coordinate data transfer operations when other scheduling tools are not available. SeeCommerce's Job Control module includes several highly desirable features that can be applied as needed:

- **Scheduling** - This allows defined data sets to be automatically acquired on a timed basis, with the interval defined in terms of minutes, hours, days, weeks, or months.
- **Event Recognition** - Jobs can be triggered based on occurrence of an event. For instance, "End of Month Reconciliation Complete" might be an event that triggers a sequence of data transfers so applications and analysis can be run at the end of the month.
- **Asynchronous Transfer Recognition** - If an external data provider such as an ETL tool asynchronously delivers data, job control mechanisms can recognize receipt of the data. It can then trigger other data transfer steps, or allow defined analysis steps to continue.
- **Multiple Steps** - A job may internally have multiple steps to be executed.
- **Sequencing** - This allows dependent data sets to be acquired in a dependent order. When one data set completes, it can trigger or allow other data transfers.
- **Parallel Execution** – Multiple jobs can run in parallel, and multiple steps within a job can run in parallel.
- **Notification** - Job results can be communicated via database logging tables, text log files, and Email. If a supported middleware system is in use, job results can be communicated via messages.

Whenever SeeChain takes an internal action, or whenever it makes an outgoing request to an external tool, that operation is wrapped with job control conventions. For asynchronous

operations that originate in external tools, SeeCommerce provides the following special support:

- For selected ETL tools, we provide pre-defined integration definitions that implement the job control protocols.
- The SeeCommerce webMethods adaptor automatically implements job control protocols with no special effort by the user.
- The SeeCommerce JMS adaptor automatically implements job control protocols with no special effort by the user.
- SeeCommerce can capture the metadata that defines the job control requirements for each data set.

LOGGING

Logging mechanisms provide ways to log the results of a data transfer operation. The logged results can be examined by other programs, or by a user. The primary logging output is to a database table, but selected logging messages can also be sent via e-mail, text files and JMS messages.

Default SeeCommerce logging includes the following characteristics:

- Job start, job completion, summary results, and execution statistical records.
- Flagging of records that fail verification checks and detailed error descriptions are logged to separate error logs.
- Logging and keeping all data, whether good or bad, as it is processed.
- The identification of all data with its source and relative position, for easy reconciliation of errors.
- Other job control and analysis steps for using the results of those logs to determine what action to take.

SeeCommerce enhanced logging support for particular data integration techniques includes:

- Our webMethods adaptor, by default, which automatically handles all log files. Otherwise, a particular connection configured by the user can override the default handling for custom error logging.
- For selected ETL tools, pre-configured integration definitions for handling the logs.

- With formatted text files, our data import tool automatically does full handling of the logs and the input files are saved in an archive directory for all formatted text files.

VERIFICATION AND TRANSFORMATION

Verification and transformation are concerned with two general issues: the integrity of the data transfer, and the business content and consistency of the data. SeeCommerce verification of the data transfer integrity includes the following steps:

- Rules can be applied at a field level for individual data transfers. For instance, particular fields should not be blank. If needed, it's possible to define that a small number of record-level errors are acceptable, but above some threshold the data transfer as a whole can or will be discarded.
- At a higher level, dependencies are defined between transfers. For instance, transfer "A" must be complete before transfer "B" is allowed to proceed.

Depending on the type of transfer, other rules can be applied. For example with formatted text files, the process of creating or moving the files around is subject to various transmission errors. To guard against this, rules can be defined to verify that the number of lines or records in the files matches an expected number, or that the content's checksums in the file match an expected number.

In addition to data transfer integrity, the SeeCommerce framework also allows for flexible application of business rules, which serve two general purposes:

- Verification of the content of individual fields, for instance, that a field contains a valid date, or that referential integrity with other data is preserved.
- Transformation of the content of individual fields, for example, to apply conversions to a normalized currency format, or to apply a calculation function against two or more fields to produce a derived field.

SeeCommerce allows point-and-click content verification and transformation from a list of predefined functions. In addition, the verification and transformation framework is extensible, allowing custom functions of any complexity to be defined using Structured Query Language (SQL) or stored procedures.

Verification and transformations of all kinds also tie in with logging. If rules are violated, logging records are generated that describe the type of error and record the location of the violation.

SeeCommerce's Summary

SeeCommerce offers you compatibility with the three leading techniques: middleware, point-to-point and formatted text files. In addition, the SeeCommerce approach provides the method with the **scalability** to integrate the large volumes of data that are needed to describe the performance of the entire supply chain; **consistency** ensuring process repeatability and data **integrity** so the user can trust the information to be accurate, and **low latency** supporting the timeliness of data that allows the business user to make decisions according to their existing business plan.

SCALABILITY

To provide insight into the extended supply chain, data is not only collected from within the enterprise but also from your trading partners and their trading partners. Hence the amount of data that is processed to provide deep insight into the performance of the supply chain can be very large.

The SeeCommerce approach is capable of handling an extraordinary number of data records (gigabytes per update cycle) by utilizing a multi-threaded architecture, which when used in conjunction with parallel server versions of Relational DataBase Management System (RDBMS), can support large data loads in a fraction of the time and meet data scalability requirements.

CONSISTENCY AND INTEGRITY

Without confidence in the accuracy of the data, the value of the information is questionable. Data is collected from numerous sources both within and external to the enterprise. As a result, there are numerous files and formats that need to be verified and coordinated to render the right results. SeeCommerce's robust verification and job control processes ensure that the data set is complete, synchronized, and correct for an accurate picture of the supply chain.

Additionally, it is important that the process be repeatable so the business user can rely on the information being available when they need it. Through SeeCommerce's Job Control module, the complete operation can be automated, enabling consistent execution as well as trouble-free processing.

LOW LATENCY

The ability to see relevant information in time to take action is imperative in supply chain performance management. By leveraging event-based data integration, transactions can be made available when they take place, allowing the user to recognize the issue for immediate action. In many instances presenting data based on a single event is not appropriate (because single data points lack context relative to trends and aggregation is often more appropriate). In these cases, data is aggregated into appropriate intervals and quickly made available through the fast loading and aggregation processes.

Whichever approach is most appropriate, SeeCommerce provides business managers with reliable and timely information across the supply chain and IT managers with an easy, unobtrusive fit within their overall data management framework.

For More Information

For more information about how SeeCommerce solutions can help you improve your supply chain performance for maximum customer satisfaction, market share, and profitability, visit www.seecommerce.com, or call 800.255.9520 (650.213.1800 outside the U.S.).

3420 Hillview Avenue
Lobby 8
Palo Alto, California 94304
www.seecommerce.com

Phone: 650.213.1800
Toll Free: 800.255.9520
Fax: 650.812.3990
Email: info@seecommerce.com