
Next-Generation ETL vs. EAI

- ▶ *Getting Beyond the Confusion*

Contents

Overview	i
Applying the Test Results in Some Clear Boundary Lines	1
Three Categories of Application Integration	2
Data Synchronization	2
Interactive Processing	3
Multi-Step Processing	4
Next-Generation ETL Defined	5
ETL	5
Packaged Application Integration	5
Real Time	6
Defining EAI	7
What is EAI?	7
MOM: A Foundation for EAI and Next-Generation ETL	7
Putting It All Together: ETL, EAI, and MOM Technology	7
Choosing between ETL and EAI	8
Data Integration Advantage: Next-Generation ETL	8
Process Integration Advantage: EAI	12
The Bottom Line	13
How Real Time Do You Need to Be?	16
Real World Examples–BusinessObjects Data Integrator in Action	17
Conclusion: Combining Next-Generation ETL and EAI	19

Overview

CIOs continue to digest the applications they have purchased over the last few years and are working towards getting them all to function together.

If the 1990s were the years for implementing packaged enterprise applications, the first decade of the 21st century is now the time for integrating those applications. Stung by the recent recession, buyers are wary of vendor hype and they simply want to get more out of in-house applications. As a result, application integration spending is markedly up in relation to enterprise applications spending. In fact, in a Morgan Stanley CIO Survey, conducted in Q1 2002, CIOs ranked application integration as their top priority.

CIO Priorities for 2002:

- #1. Application integration
- #2. Connecting to customers over the internet
- #7. Connecting to suppliers over the internet
- #9. Business intelligence tools¹

Integration across applications provides broader access to accurate, consistent, and complete data by employees, suppliers, and customers, resulting in more efficient operations, more satisfied customers, and faster, more effective decisions. Customers get instant online access to inventory availability and order status. Planners get access to suppliers' inventory and available-to-promise data. Customer support, customer planning, and the customers themselves get a 360^o view of the customer.

As part of these integration efforts, organizations are moving larger and larger data volumes between enterprise applications, and performing complex transformations on the data in the process. This is a task well suited for extraction, transformation, and loading (ETL) technology, and challenging for pure enterprise application integration (EAI) technology. Originally designed for building data marts and data warehouses, and updating them in batch mode, the capabilities of next-generation ETL tools have been expanded to also meet the requirements of application integration. These data integration tools are combining batch ETL, elements of real time, and packaged data movement across enterprise applications to provide capabilities once considered to be the reserve of EAI tools. Features such as bi-directional packaged application interfaces, guaranteed delivery, and even real-time data movement are now key components of these tools.

Despite these areas of overlap, next-generation ETL tools for application integration remain complementary to EAI technology—the technology most commonly applied to application integration challenges. However, as these tools progress beyond core batch data warehousing, the choice as to when to use EAI or ETL has become increasingly confusing. The line has begun to blur between EAI and ETL technologies.

This paper outlines the strengths and weaknesses of each technology, and draws clear boundaries around the types of application integration projects most appropriate for each technology. It is no longer as simple as ETL for batch/bulk data movement and EAI for real time. While you will still typically use EAI technology more often than ETL for real-time, application-to-application integration, it is important to note that next-generation ETL tools can also handle real-time data movement and are helping organizations solve more complex business problems.

¹ Source: "Morgan Stanley CIO Series: Release 3.1," March 21, 2002.

This paper will compare and contrast data integration with process integration—and will explain why next-generation ETL tools are most appropriate for data integration while EAI tools are best for process integration.

A gray area, called interactive processing, sits between data integration and process integration. Interactive processing involves executing a transaction that is split across two or more applications, but requires complete continuous processing with no workflow interruptions that require human intervention and discontinuous processing. In these cases, either EAI or ETL technology could be applied. A two-part litmus test that measures productivity and performance will be outlined to help determine which technology to use for interactive processing. Maximizing developer productivity for a particular integration project requires determining which tool's graphical user interface (GUI) development environment enables you to do the job without having to drop down into hand writing code. The second part of the test involves determining which tool automatically provides maximum performance.

Applying the Test Results in Some Clear Boundary Lines

ETL tools are most appropriate for data integration that consists of data synchronization between applications, and for point-to-point, single step interactive processing. Real-time data oriented integration projects that involve large amounts of data, complex transformations, or data augmentation are appropriate for these tools. In these cases, you can typically design your entire integration job within the GUI of the ETL tool, while in many cases you'd have to drop down into coding when using an EAI tool. You will also get better performance moving and transforming large chunks of data with an ETL tool performing relational database-type operations on large amounts of data.

EAI tools are clearly most appropriate for process integration, which consists of multi-step business process management and real-time interactive processing when very large numbers of transactions are involved. ETL tools do not handle these processes well. ETL tools are not designed to handle discontinuous workflows, or to scale to moving very large numbers of small transactional messages.

■ *Figure 1: ETL for data integration; EAI for process integration*

Data Integration ETL		Process Integration EAI
Data synchronization ETL Batch and real-time application data synchronization	Interactive processing (ETL or EAI) Point-to-point Continuous processing Simple or no workflow	Multi-step processing EAI Workflow BPM Multi-step process

If you understand that next-generation ETL technology uses some of the same technology as EAI to provide real-time application integration, it will help you recognize the difference between EAI and ETL. The basic underlying technology for EAI, called Message Oriented Middleware (MOM), uses store-and-forward queuing technology to provide guaranteed delivery of messages. Both next-generation ETL and EAI technologies leverage MOM. EAI builds workflow and process integration on top of MOM. ETL builds its real-time data integration on top of MOM. Because it's critical that you understand differences between EAI and ETL, this paper will explain the distinct differences between what EAI and ETL technologies provide on top of MOM.

Three Categories of Application Integration

Analysts generally list three major categories of integration patterns:

■ *Figure 2: Three categories of integration*

Data Synchronization	Interactive Processing	Multi-step Processing
The same data needs to be in two or more systems	A transaction needs to be completed across systems	A business process where a number of transactions occur in steps through a pre-defined sequence across two or more systems
Getting two or more systems to agree on the facts	Synchronous interactions among closely knit participating systems	Multi-step processes tie systems together in an asynchronous series of steps with various dependencies
Batch and real-time	Also called Straight-through processing and composite applications	
Data migration to seed new apps		

► Data Synchronization

Data synchronization involves initially seeding historical data into new applications in a batch load operation, and ongoing synchronization of the data—some in batch and some in real time. Integration of the data is needed across ERP, CRM, SCM, and other enterprise applications. For example, orders entered into the ERP system have to be shared with CRM systems so that customer service representatives, distributors, and customers who are accessing a corporate BI portal have complete and timely information. Orders entered into the CRM system have to be synchronized across to the ERP system. Orders also have to be transferred to manufacturing systems for execution. The data has to be cleaned, consolidated, and transformed along the way.

As with data warehousing, much of the data synchronization can be performed in batch. Master reference data, such as customer type, history, and preferred shipping methods can typically be updated once a night, or even once a week. Critical data on order status, new customers, and inventory availability increasingly needs to be updated in real time. Thus, data synchronization requires a combination of batch and real-time updates. Updating everything in real time is not only unnecessary, but may require building custom interfaces or APIs. It also puts an undue burden on the developer to develop real-time flows and overloads operational systems with unnecessary data movement during peak operational hours. Most organizations find the need to perform a lot of batch data integration and a moderate, but growing, amount of real-time data integration for data synchronization.

Examples of data synchronization:

- Pulling orders from your ERP system and updating your CRM system so that your telesales operators have up-to-the-minute information
- Synchronizing or consolidating customer information across systems so that you get a unified 360^o view of your customer
- Populating an operational data store in real time so that your customers and/or distributors can view inventory availability and order status via a business intelligence (BI) extranet
- Pulling data from your ERP system and loading it into your SCM system once or several times a day in order to do demand planning
- Shipping pricing information multiple times a day to distribution channels
- Shipping exchange rate information to worldwide subsidiaries multiple times a day

► **Interactive Processing**

The second type of application integration is interactive processing. This involves executing a transaction that is completed across two applications. This processing is complete and continuous and does not involve any workflow that requires human intervention or discontinuous processing. Also, because this process is usually between two applications, it does not require the typically complex routing of EAI.

Examples:

- Transferring orders from your ERP system to your shop floor systems for picking, packing, and shipping
- Transferring distributor and customer orders into the ERP system from a web-based front-end order taking portal

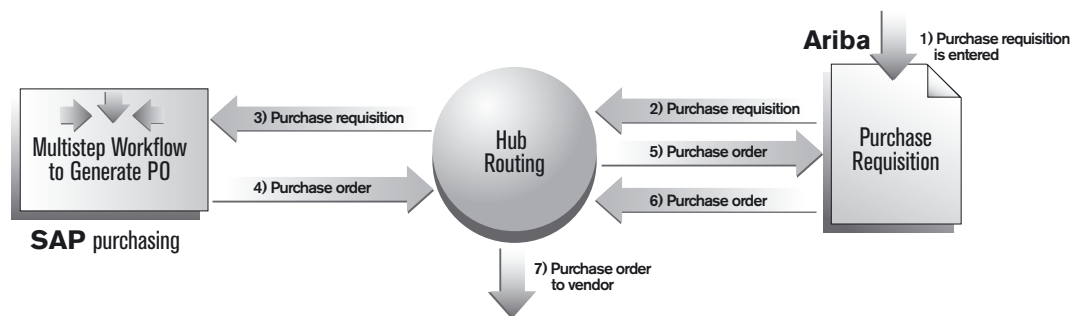
► Multi-Step Processing

As part of a business process, a number of individual transactions occur in steps through a predefined sequence across two or more systems. This is also referred to as workflow or Business Process Management (BPM). The process involves a series of steps and many systems, and can take an hour, days, or even weeks to complete. It can be 1-to-1, 1-to-N, N-to-1, or M-to-N.

Examples:

- Automated online order entry, order validation, financial approval, and shipping
- Purchase order approval and execution

■ *Figure 3:
Sample Workflow*



A purchase requisition is created in Ariba, but needs to go through a multi-step approval process, be entered as a purchase order into SAP, and only then can it be sent to the vendor.

Next-Generation ETL Defined

Next-Generation ETL = ETL + Packaged Application Integration + Real time

“An ETL tool is data integration software that facilitates extraction of data from multiple data sources. Using business rules, the data is integrated and transformed in preparation for loading to a target data warehouse, data mart, or other application database. Most ETL tools can access a range of data sources and target types (data formats), include a library of built-in transformation functions, and provide some degree of support for the operational aspects of data movement (e.g., scheduling, job control, and error handling).”²

▶ ETL

Traditional ETL tools provide graphical drag-and-drop user interfaces to design data movement from client-server and other enterprise applications to data warehouses. Most ETL tools automatically generate SQL to extract data from relational databases. However, as demand has grown for real-time analytics and for hooking together enterprise applications in batch and real time, data integration vendors have had to extend their ETL tools to include sophisticated access to the application metadata of leading enterprise packaged applications, as well as the ability to leverage their real-time interfaces.

▶ Packaged Application Integration

Next-generation ETL tools extract data and business logic from packaged enterprise applications via the application layer using packaged interfaces. With ERP, SCM, and CRM applications like Oracle eBusiness Suite, PeopleSoft, and SAP R/3, much of the business logic required to understand the data stored in the underlying relational database has been built into the application layer of the applications. Directly accessing the underlying DBMS causes problems and generating traditional SQL is simply not enough. A specific way to extract an application’s business logic is needed, and so next-generation ETL tools were born. These tools can provide tight integration with leading packaged enterprise applications.

Next-generation ETL tools interact with and understand the application layer to ensure that all the business meaning of the extracted data is captured. Working via the application layer means that in addition to generating SQL, these tools work closely with the data dictionaries and repositories of the enterprise applications to understand the meaning of the data. Application interfaces that hook to and read from the application’s dictionary and present logical source data in a simplified and standard form to the ETL developer are required. These interfaces also generate code specific to the applications and deal with data APIs and data structures unique to each application. For SAP, they generate ABAP and call RFCs. For the Oracle eBusiness Suite, they work with the data dictionary to understand Flexfields. For PeopleSoft, they can traverse effective dates, domains, and a variety of encoded hierarchies. For J.D. Edwards, they must convert date and floating-point data from proprietary application formats. In addition, next-generation ETL tools provide not only a wide array of out-of-the-box interfaces and transforms, but can be delivered with prebuilt data integration jobs for rapid deployment, such as for SAP-to-Siebel integration.

² Source: “Integration Brokers and ETL Tools: Is the Line Blurring?” Gartner. November 14, 2001.

► Real Time

Next-generation ETL tools move data in real time. They incorporate the following capabilities:

■ *Figure 4:*
Next-generation ETL
real-time requirements

1. **Real-time message processing server:** The ability to process incoming messages and trigger outgoing messages in real time from any application. The key to a real-time message processing system is the set of components that continuously listen for requests to process.
2. **Real-time data flows:** The ability to graphically design real-time data flows. A real-time data flow includes logic to pull data from ERP and other enterprise systems, to supplement a request, and to construct a reply. The real-time data flows process requests in the form of XML messages created by web clients, such as eCommerce applications, and also return responses as XML messages.
3. **Administration capabilities:** Web-based administration for the full lifecycle management of real-time interfaces across the enterprise—configuration, starting, stopping, and status monitoring.
4. **Complex structural transformations of hierarchical data from within the GUI:** The ETL tool must be able to easily transform hierarchical documents, such as XML or EDI documents, to a relational format, and to operate on the hierarchical structures without the need for the developer to transform them into a relational structure first. Having to break the data down into a flat format is cumbersome for a developer, often causing some loss in the meaning and context of the data. It also degrades performance. Hand coding these transformations would be highly complex and difficult. Next-generation ETL tools embody the ability to deal with transformations on an NRDM (Nested Relational Data Model) from within the GUI, without having to hand code.
5. **Batch and real-time data flows in one tool:** The ability to share common data definitions to ensure data consistency across batch and real-time processes.
6. **Bi-directional real-time interfaces:** Real-time metadata integration is required with a wide array of tools and applications:
 - ERP, CRM, and SCM application real-time interfaces (such as SAP IDocs and Oracle Triggers)
 - Enterprise servers (via J2EE, JCA, JMS, and HTTP)
 - Web services (support for SOAP, WSDL, and UDDI)
 - BI tools (via HTTP by parsing XML documents)
7. **Interfaces for leading EAI/MOM:** Interfaces are required for message oriented middleware (MOM) software for guaranteed delivery (e.g., TIBCO Rendezvous/TIB and IBM WebSphere MQ).
8. **Real-time interface framework:** A next-generation ETL tool should provide a messaging infrastructure and interface framework that enables rapid building of native interfaces to any application or tool where out-of-the-box adapters/interfaces are not available. A typical framework provides a set of modifiable Java Class Libraries, with defined APIs and a fully documented implementation methodology for handling the full lifecycle management of the interface—configuration, starting, stopping, and status.

► What is EAI?

The leading EAI tools include graphical development tools for defining routing flows, transformation rules, and security. They provide off-the-shelf adapters for packaged applications and adapter development tools. Evaluation criteria for EAI often include ease of use and power of the development tools, throughput, scalability, reliability, administration, and management. Transformation capabilities focus on syntactic conversion and semantic transformation for XML and other data types. They provide their own MOM, in addition to gateways to external platform middleware and MOM products.

“An integration broker (EAI tool) is a software intermediary (hence, ‘broker’) that facilitates interactions among application systems. A broker supports transformation of messages, files, or calling parameters, and intelligent routing (e.g., content-based routing or publish-and-subscribe). Most integration broker suites also offer business process management (BPM) and adapters to packaged applications and heterogeneous software platforms.”³

MOM: A Foundation for EAI and Next-Generation ETL

As mentioned in the overview, both next-generation ETL and EAI tools build on some of the same underlying technology to provide real-time capabilities—message oriented middleware, or MOM. A very simple definition for MOM is that it provides guaranteed once-only message delivery. You provide a message to MOM, it places it in a message queue, and then the MOM ensures it gets where it’s going.

Putting It All Together: ETL, EAI, and MOM Technology

Performing data synchronization, interactive processing, and multi-step processing requires a mix of all the technologies discussed so far—ETL, EAI, and MOM.

MOM provides both EAI and ETL tools with guaranteed delivery, in addition to other capabilities such as publish, subscribe, or broadcast. The difference is the graphical application built on top of MOM:

- EAI workflow products provide graphical development and management of workflow and BPM on top of MOM
- EAI uses MOM for interactive processing and multi-step processing, most distinctively when involving large numbers of transactions or when complex distribution one-to-many or many-to-many distribution is required
- Next-generation ETL uses MOM for guaranteed delivery for real-time data synchronization and interactive processing
- Next-generation ETL on its own handles batch data synchronization and certain real-time interactive processing scenarios (plus tasks traditionally handled by ETL such as batch and real-time data warehousing)

³ Source: “Integration Brokers and ETL Tools: Is the Line Blurring?” Gartner. November 14, 2001.

■ *Figure 5:*
The technology stacks—
technology requirements/
underpinnings

Data Synchronization	Interactive Processing		Multi-step Processing	
			BPM	
			Workflow	
ETL (batch + real time + packaged integration)	ETL (batch + real time + packaged integration)	EAI	EAI	GUI design and admin Intelligent routing transforms adapters
	MOM	MOM	MOM	Transport, publish and Subscribe store and forward Guaranteed fault tolerance

- Next-generation ETL tools are best for data synchronization
- EAI tools are best for multi-step processing
- Either EAI or ETL tools can be used for interactive processing

To determine which technology to use for a particular integration problem, we use a two-part litmus test that measures both productivity and performance:

1. Determine which tool, ETL or EAI, can do the complete development job within the tool's GUI development environment without having to drop down into hand writing code.
2. Determine which tool will provide better performance.

Choosing between ETL and EAI

► Data Integration Advantage: Next-Generation ETL

Next-generation ETL is clearly the right technology for data integration, whether in batch or real time. Synchronizing data between two applications involves a lot more data manipulation than simply moving data from point A to B; there's reconciliation, cross matching, de-duping, and cleansing. These are all data intense tasks that depend upon either RDBMS efficiencies/scalability or in-memory data caching to achieve the necessary throughput. Typically, enterprise data warehousing projects require you to move large amounts of data within relatively small windows of time. Performance therefore plays a critical role. The more data you need to move, the more complex the data manipulation, the more likely a proven, next-generation ETL tool is appropriate.

ETL tools were born out of the relational database world, and thus are adept at performing SQL-oriented transformations on sets of relational data. They are oriented towards pulling data out of multiple relational tables, understanding the meaning and relationships between the tables, combining, merging, or joining that data, and augmenting it with data from other sources. This may involve simple joining of two relational tables, or complex heterogeneous joins involving multiple tables from different applications. It can also involve very complex transformations. Next-generation ETL tools enable the design of very complex, set-oriented extractions and transformations via the GUI, without having to write a single line of code. They automatically generate the appropriate optimized SQL code, or the appropriate optimized code for the packaged application (e.g., ABAP for SAP R/3).

Think of a next-generation ETL tool as providing a graphical front-end for doing database joins and operations. It offers a graphical representation of what an RDBMS can do—what SQL can do—at both the simple and very complex levels. Let's take the example of executing a decode or a lookup for a particular RDBMS or for SAP R/3. If you had to write code then you would have to worry about the syntax for the function for the appropriate language—SQL for the RDBMS and ABAP for SAP—whereas with the right ETL tool you simply fill out a form and the right code is automatically generated. Plus, the ETL tool will automatically optimize the operation. Even a relatively simple and obvious thing to ETL tools, such as the order of joins, has to be handled manually by EAI tools.

As the transformations get complex, the relative strength of a next-generation ETL tool over an EAI tool grows. EAI is message oriented, not data set oriented. So, if you need to take a data set, sort it, pivot it, flatten the hierarchy, and write out the result set, you would have to write and optimize a lot of code with an EAI tool. On the other hand, this is a typical transformation done and optimized with the ETL tool by simply filling out a form.

ETL tools are also more suited for set-oriented data processing. Ultimately, most of the data will reside in an RDBMS that is inherently more scalable when asked to return a range of data (for example, "all open orders associated with customers from California that are new or have changed since the last update") than with multiple single-record function calls. For extractions and transformations of large amounts of data, you need to focus on the changed-data interface that will provide for the greatest SELECT efficiencies by utilizing some highly selective, and thus efficient, WHERE predicates/clauses and not a series of API calls. A good example is to flatten or reconstruct organizational, sales, or accounting hierarchies, which would be very difficult without access to all the data.

Real-time interactive processing that involves data augmentation, not just batch data synchronization, is also appropriate for next-generation ETL tools, particularly if heterogeneous joins are required to integrate data from multiple applications. For example, transformations are required if an order being transferred from an ERP to a shop floor system has to be augmented with master data describing the customer's preferred shipping method, credit status, or priority rating. These transformations are DBMS operations—well handled by the GUI of an ETL tool.

Even with hierarchical data, such as XML, an ETL tool is well suited for complex transformations. While EAI tools are known to handle XML, it's up to the user to define the content. EAI tools can send and receive XML, but do not handle unpacking the data, understanding it, and transforming or augmenting the data as well as an ETL tool. With an EAI tool, it's typically up to the applications themselves to perform those operations before sending or receiving the data. An ETL tool's nested relational model capabilities allow the developer to use the GUI to graphically navigate the hierarchical XML structures to perform operations such as identifying individual orders in a structure with multiple order line items per header, augmenting that information with relational operations, and then sending the augmented message to the downstream application.

EAI tools are not generally designed to understand the data schemas of the applications and to perform data transformations. They are designed to interact with the applications at an API level. APIs are most commonly defined for specific transactional integration, and not for enabling broad integration of any data or sets of data in the application. If no API exists for accessing the data you require, you must write your own, which means hand coding. Furthermore, you typically have to drop down into writing code in order to perform and optimize data transformations. Consequently, it is more complex, time consuming, and difficult to use EAI tools to communicate and share data. For example, if you customized your ERP system by adding new attributes to a certain object (e.g., customer), the packaged APIs that access that object would also have to be modified for you if you were relying upon EAI tools for this task. Even if you used an EAI tool to write the API and transformations by hand, it is likely that performance will be significantly better with an ETL tool because it has been designed from scratch to maximize extraction and transformation performance. As well, next-generation ETL tools include performance optimization techniques such as those listed in Figure 6.

■ *Figure 6:*
Next-generation ETL
optimization techniques

Automatic workload distribution: The ability to put the ETL work where it is most efficiently executed—at the source, target, or in the ETL engine. The ETL tool should automatically push down operations to source and/or target engines, thereby enabling load balancing among source, engine, and target servers. For example, you may wish to push down a sub string or aggregation operation rather than pull all the data out of the DBMS before you perform the required transformations.

Intelligent threading: The ETL tool should automatically break each data flow into separate components and launch each component as a separate operating system thread, thereby utilizing the multi-threading power of the operating system to maximize the resource utilization of multi-processor systems.

Parallel and distributed data flow execution through parallel pipelining: The ETL tool should provide sophisticated automatic parallelization. The mappings and transformations specified with the ETL tool's GUI are parsed and individual operations are identified. Typical operations include reading a row of data from a source table, calculating a sum, formatting the data in a column, performing a lookup, generating a key for a dimension table, or writing bad data to an error file. Each operation is then executed on a separate thread and the data streams through in "assembly-line" style. For example, instead of waiting for all of the rows in a table to be read before applying data transformations, one thread in the system reads the table row-by-row, and another thread operates on each row of data as it is read. Data streaming allows all of the operations in the sequence to work in parallel with less need for storing the interim results in the process. ETL tools enable multiple instances of the ETL engine to be launched to run each operation in parallel, either within one server or on multiple servers.

Integration with high-speed DBMS bulk loaders: Native access to high-performance load utilities from multiple RDBMS vendors, in a declarative fashion (e.g., by just filling out a form).

Parameterized database SQL loading: Precompiled SQL to speed up database loads.

In-memory caching: For most operations with no need for intermediate staging of data between transformations.

It is worth noting that when a next-generation ETL tool is combined with MOM for guaranteed delivery this is mainly driven to account for poor LAN or line connectivity, and there is generally no significant performance impact for moving large amounts of data. Thus, even when you combine ETL with MOM as the transport mechanism, the ETL tool still provides significant performance benefits.

► Process Integration Advantage: EAI

EAI tools are appropriate for process integration—moving and tracking documents through stages. Process-level EAI deals with building enterprise-wide business workflows and processes and incorporating existing applications into those processes. EAI middleware acts as the workflow engine integrating applications in near real time, passing small amounts of data through message queues and a series of stages. EAI tools provide much more complete workflow capabilities than ETL tools, which provide simple workflow. EAI tools, and especially their workflow components, provide very sophisticated GUI development environments that enable design and management of very complex business processes.

Like ETL tools, EAI tools enable transformations. In fact, the leading EAI tools have robust libraries of transformations. However, the types of transformations enabled are generally of a different nature than those enabled by an ETL tool. EAI tools grew up out of the need to move individual transactions. Consequently, typical EAI transformations perform rules-based data transformation and validation to resolve differences between data fields, models or import/export formats. Many EAI transformations are focused on ensuring a common understanding of the context and meaning (semantics) of the data involved within the message. Most are designed to work on single rows of data. They are typically not designed for working on sets of data, and are therefore not geared towards the data transformations and augmentations that are typically performed by ETL tools.

■ *Figure 7:
Typical ETL vs. EAI
transformations*

ETL (Complex transformations)	EAI (Elementary or process transformations)
<p><u>Data Set Oriented</u></p> <ul style="list-style-type: none"> ■ Aggregation ■ Heterogeneous joins ■ Hierarchy flattening ■ Data augmentation ■ History preserving ■ Effective dates ■ Table comparisons (for history preservation) ■ Merge ■ Pivot (Convert rows to columns or columns to rows (e.g., for hierarchy flattening)) 	<p><u>Act on a single row of data</u></p> <ul style="list-style-type: none"> ■ Semantic functions/syntactic functions ■ String functions ■ Substring ■ Trim ■ Concatenate ■ Date conversions/functions ■ Year ■ Month ■ Tochar ■ Math functions ■ Truncate ■ Round <p><u>Process Oriented</u></p> <p>Navigation through a document structure— one document at a time.</p>

EAI tools provide better performance than ETL tools for moving large numbers of individual messages or transactions, especially if they are moved from one-to-many locations. For the last decade, EAI tools have focused on providing highly scalable one-to-many, and many-to-many, real-time transactional message distribution and queuing. EAI tools have evolved to handle scenarios that involve millions of transactions per hour. They offer robust capabilities to distribute and parallelize workflow components to run on multiple servers, and to handle difficult situations such as when one or several of the servers go down. They are adept at distributing transactions and components across resources.

► The Bottom Line

ETL tools focus on integrating data. EAI tools handle processes.

As we've explained, next-generation ETL tools are most appropriate, as compared to an EAI solution, for batch or real-time data synchronization between applications where a large amount of data is being extracted from an application and the data has to be transformed (typically with SQL or XML type transformations), and then loaded into another application. EAI solutions are more appropriate where workflow and business process management is required, which typically involves moving a large number of small transactional messages through an approval process, and little data transformation is required.

For interactive processing, if no extensive workflow is required, or complex data transformations are required, or a combination of batch and real-time data flows are required, an ETL tool is most likely a more efficient and effective tool.

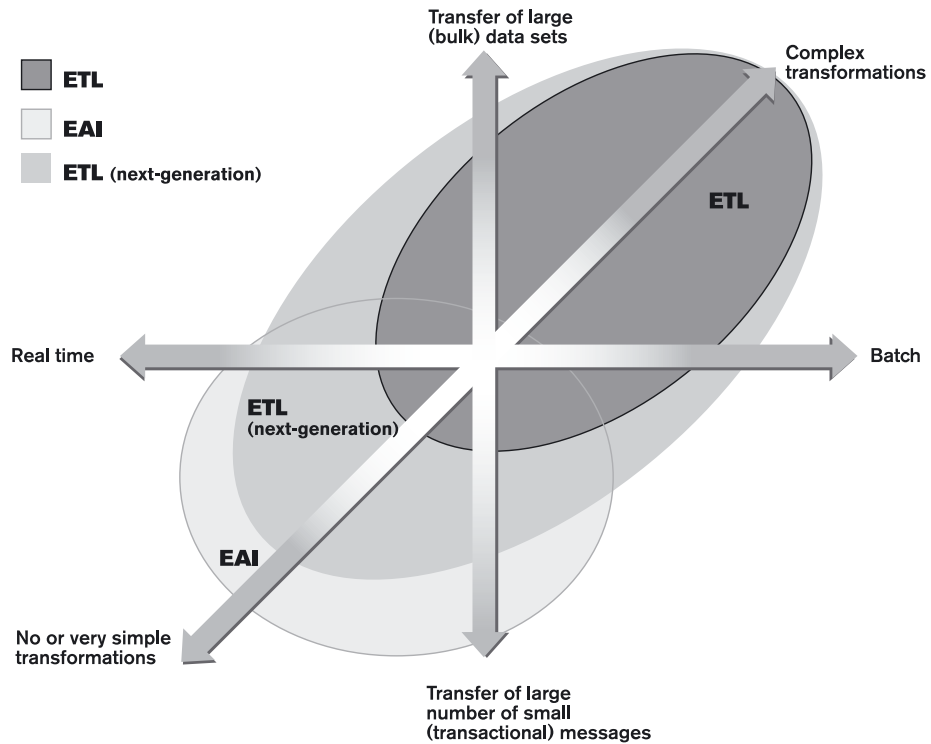
■ *Figure 8:*
Next-generation
ETL vs. EAI

Data Integration ETL		Process Integration EAI	
Data synchronization ETL Batch and real-time application data synchronization	Interactive processing (ETL or EAI) Point-to-point Continuous processing Simple or no workflow	Multi-step processing EAI Workflow BPM Multi-step process	
<u>Advantage ETL when:</u> Large amounts of data Use ETL already Complex transactions Lots of data augmenting Point-to-point Manufacturing		<u>Advantage EAI when:</u> High # of transactions Use EAI already In message transforms Little data augmenting 1-to-n; m-to-n Wall Street	

■ *Figure 9:*
Next-generation
ETL vs. EAI for
interactive processing and
data synchronization

	Advantage EAI	Advantage ETL
Huge volumes of transactions	X	
One-to-many; many-to-many	X	
Complex transformations/data augmentation		X
Large amounts of data		X

■ *Figure 10:*
EAI vs. Next-generation
ETL for interactive
processing and data
synchronization³



³ Source: Modified version of diagram by Philip Russom, "Beneath the Waterline," Intelligent Enterprise, May 7, 2001.

■ Figure 11:
EAI vs. ETL

ETL (next-generation)	USES	EAI	Description	Why ETL or EAI?
NO	Multi-step (BPM/Workflow)	YES	Coordinates high-level business processes inside your company and across supply and distribution channels.	Requires sophisticated workflow management provided by EAI.
YES	Interactive Processing	YES	Transactions integration. Continuous, synchronous transaction execution between apps with minimal workflow. Composite apps. Straight-through processing. Point-to-point or point-to-many points.	EAI if little/no transformations or very large # of transactions. next-generation ETL if large amounts of data, complex transformations or data augmentation.
YES	Real-time Data Synchronization	YES	Real-time, event-driven data synchronization between applications. Point-to-point or point-to-many points.	Requirements similar to next-generation ETL requirements for updating a real-time ODS. Large amounts of data, data transformations or data augmentation.
YES	Batch Data Synchronization	NO	Batch data synchronization between applications. Also, initial batch loading of a new application with data from a legacy app.	Similar requirements to batch data warehousing. Requires complex transformations and high performance for moving large amounts of data provided by next-generation ETL.
YES	Operational Data Store	NO	Real-time feeding of detailed updates to a data warehouse.	Requires both real-time data flow and extensive data warehousing transformations only provided by next-generation ETL.
YES	Data Warehouse	NO	Batch weekly, daily or multiple times a day updates to business intelligence DB from multiple sources for multiple subjects	Batch extraction and complex set oriented transformations and movement of large amounts of data only provided by next-generation ETL.
YES	Data Mart	NO	Ensures consistent definitions and data content across multiple enterprise apps and data warehouses, such as customer, product, geographic definitions, hierarchies (batch or real time).	Requires deep understanding of the data and metadata, combined batch and real-time data movements, and very complex heterogeneous transformations.
YES	Master Data Synchronization	NO	Ensures consistent definitions and data content across multiple enterprise apps and data warehouses, such as customer, product, geographic definitions, hierarchies (batch or real time).	Requires deep understanding of the data and metadata, combined batch and real-time data movements, and very complex heterogeneous transformations.

■ Proceed with caution

How Real Time Should You Be?

Prior to determining whether to use EAI or ETL technology, organizations need to determine their true real-time requirements, and to do it separately for each integration project. The answer for most companies will be a lot of batch and a moderate, but growing, amount of real-time integration. Performing all integration in real time may seem desirable, but justification usually lacks clear business drivers. It places an unnecessary burden on developers to create and maintain real-time flows, and on operational systems to perform unnecessary data movement during peak operational hours.

- A disk drive manufacturer who ships one million disk drives per week updates its data warehouse three times a day with order, inventory, and other critical information required to maintain maximum flexibility to adjust manufacturing, order fulfillment, and shipping plans multiple times a day. Simultaneously, it also requires real-time, event-driven updates on a limited subset of the data to an operational data store so that distributors can get up-to-the-minute information on order status and inventory availability.
- A European gasoline manufacturer and distributor transfers orders from its ERP to its fulfillment software every 10 minutes.
- A plastics manufacturer sends orders from its ERP system to its shop floor system on an event-driven, real-time basis, but finds that batch refreshes from the shop floor back to the ERP with planned delivery dates and times is sufficient.
- A Wall Street brokerage requires event-driven, sub-second updates of trading data across a wide array of applications.

After determining real-time requirements and evaluating EAI vs. ETL technology, many organizations will conclude the need for both. No single technology solves all integration tasks today. Let's look at some examples of situations that may require both complementary technologies.

Real World Examples— BusinessObjects Data Integrator in Action

Let's now take a look at several real world examples where companies made a choice between ETL and EAI.

For the examples listed, the data integration product of choice is BusinessObjects Data Integrator.™ Data Integrator is the industry's first real-time and batch data integration platform to effectively and intelligently share data between all enterprise data sources. Data Integrator expands ETL technology beyond the traditional data warehousing role and is now the global data integration standard for corporations with the most demanding requirements for power, speed, and flexibility. A powerful next-generation ETL tool with industry-leading performance and a proven track record of providing a quick ROI to Global 2000 enterprises, Data Integrator can be delivered with a wide array of packaged out-of-the-box interfaces, transforms, and even complete data integration jobs for rapid deployment.

■ *Figure 12:*
Examples—
Next-generation
ETL and EAI
in action

Integration type	Technology used	Industry	Description
BPM/workflow	EAI (real time)	Retail	Control the flow of goods to 4000 stores and conduct transactions with delivery agents through secure exchange of documents. EAI used for complex workflow design.
Interactive processing	EAI (real time)	Semi-conductor manufacturer	Real-time integration between tracking software, statistical process control techniques and automated handling devices. EAI used for many-to-many capability.
Interactive processing	ETL+EAI/MOM (real time and batch)	European-wide retail operation	Download price data and upload transaction data between stores and ERP. Data Integrator used for easy hierarchical transformations and data augmentation and EAI used for guaranteed delivery.
Interactive processing	ETL (real time and batch)	Plastics manufacturer	Real time from ERP to shop floor for picking, packing, and shipping with data augmented from other systems. Data Integrator used for ease of augmentation and transformation.
Near real-time data synchronization	ETL (near real time, frequent batch)	Gasoline production and distribution	Feeds orders from ERP system to distribution software every ten minutes. Data Integrator used plus batch ERP interfaces.
Batch data synchronization	ETL (batch)	Manufacturing (Leading SCM software vendor)	Large amounts of batch data moved and transformed into supply chain planning application several times a day. Data Integrator used for performance and ease of handling complex transformations.
Real-time ODS	ETL + EAI/MOM (real time)	Energy trading	EAI feeds for trade executions passed to Data Integrator for transformation and real-time and daily updating of data warehouse
Real-time ODS	ETL (real time)	Chemicals	Data Integrator updates order status from ERP system to ODS in real time to support customer portal
Data warehouse	ETL (batch)	CPG	Data Integrator performs batch extractions from 20 different enterprise applications into a central data warehouse for analytics.

Conclusion: Combining Next-Generation ETL and EAI

According to Gartner, “More than 80 percent of companies who lead their respective industries in revenue growth during 2002 to 2004 will have implemented a real-time enterprise nervous systems (ENS) for integrating applications within and outside the enterprise.”⁴

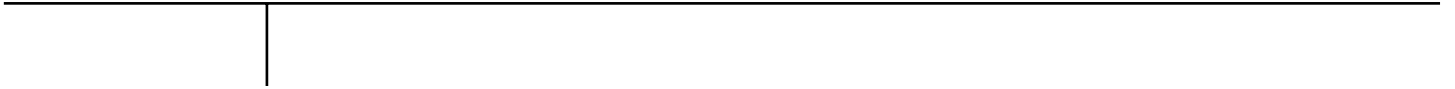
Gartner states that more than half of all enterprise nervous systems will leverage integration broker (e.g., EAI) suites as enabling technology. Gartner also concludes in a separate, but related, paper that companies will need both ETL and information brokers (i.e., EAI). Business Objects agrees.

As for when to use EAI and when to use ETL, Gartner essentially boils it down to bulk data movement versus real time—ETL for bulk, information brokers (i.e., EAI) for real time. Again, Business Objects agrees with this batch vs. real time conclusion as it applies to the bulk of ETL vendors-but also believes that bringing next-generation ETL technology into the equation modifies the conclusion somewhat such that enterprise nervous systems for many organizations will rely on a combination of next-generation ETL and EAI technologies.

Next-generation ETL solutions enhance integration productivity and performance for the batch and real-time data integration tasks of data synchronization and interactive processing above and beyond what EAI tools offer.

For more information on the data integration products from Business Objects, please visit www.businessobjects.com/products/data_integration/

⁴ Source: “The Enterprise Nervous System Arrives,” Gartner, December, 2001.



**Americas**

Business Objects Americas
3030 Orchard Parkway
San Jose, California 95134
USA
Tel: +1 408 953 6000
+1 800 877 2340

Asia-Pacific

Business Objects Asia Pacific Pte Ltd
350 Orchard Road
#20-04/06 Shaw House
238868
Singapore
Tel: +65 6887 4228

Europe, Middle East, Africa

Business Objects, SA
157-159 rue Anatole France
92309 Levallois-Perret Cedex
France
Tel: +33 1 41 25 21 21

Japan

Business Objects Japan K.K.
Head Office
Yebisu Garden Place Tower 28th Floor
4-20-3 Ebisu, Shibuya-ku
Tokyo 150-6028
Tel: +81 3 5720 3570

For a complete listing of our sales offices, please visit our website.

► www.businessobjects.com