

Analyze and Act on Fast-Moving Data

An Overview of Complex Event Processing



Table of Contents

- 5 **What Is Complex Event Processing?**
 - An Alternative to Custom Code
 - A New Concept and New Terms
 - A Spectrum of Uses and Requirements
 - Real-Time Data Analysis

- 8 **SAP Event Stream Processor**
 - Designed for Performance
 - Designed for Versatility
 - Designed for Productivity
 - System Integration
 - SAP Event Stream Processor Product Features

- 13 **Typical Complex Event Processing Applications**
 - Take Advantage of Next-Generation Technology

Learn the underlying concepts and benefits that can be gained by using complex event processing technology to address the high-performance needs of today's real-time enterprise.



In the capital markets, things happen fast. The markets move fast. Positions change fast. **Reacting quickly** is the key to increasing profit and managing risk – whether it’s an automated trading or market-making application or a risk manager who wants to see the firm’s exposure updated continuously throughout the day. It all comes down to the ability to analyze data arriving from multiple sources, at very high rates, in real time.

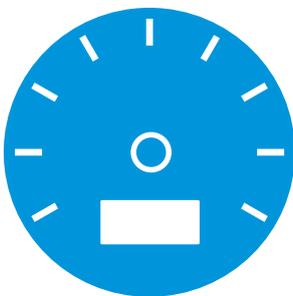
This phenomenon is by no means exclusive to the capital markets. In addition to financial services, many industries – telecommunications and networking, logistics and transportation, and government, to name a few – are experiencing the same challenges. These include growth in data volumes, acceleration of rates at which data is received or created, and increasing business pressure to analyze the data and act on it in real time.

Complex event processing (CEP) technology provides an innovative approach to deriving intelligence from event data in real time. As a platform for application development, CEP provides high-level tools for defining how events will be processed and

analyzed. As an engine for an event-driven architecture (EDA), it provides the “brains” to absorb, aggregate, correlate, and analyze events. These events produce new high-level events that can trigger a response, as well as high-level information that shows the current state of the business. CEP lets you easily define logic that will be applied to incoming events (that is, messages) to do things such as:

- Combine data from multiple sources, producing derived streams with richer and more complete information
- Compute value-added information to enable rapid decision making
- Watch for specific conditions or patterns to enable instantaneous response
- Produce high-level information, such as summary data, statistics, and trends to be able to see the big picture, or the net effect, of many individual events
- Continuously recompute key operating values based on complex analysis of incoming data
- Collect raw or result data into a historical database for historical analysis and compliance

This paper introduces you to the concepts underlying complex event processing and the benefits that can be gained by building event processing applications using SAP® Event Stream Processor (SAP ESP) – the most complete enterprise-level CEP technology available for today’s demanding requirements.



Complex event processing technology delivers the data analysis tools traditionally provided by relational databases. But CEP delivers those tools in a **real-time, event-driven implementation** capable of processing incoming data at very high rates.

What Is Complex Event Processing?

Consider the following situations:

- An automated trading application that scans massive amounts of incoming market data to spot trading opportunities, where the trigger to trade has to be instantaneous, or the opportunity is missed
- A market-making application that has to adjust internal or published rates in response to market movements, where delays either mean lost business or lost profit
- A risk management application that continuously updates aggregate position and risk information, combining data from multiple systems to provide a single consolidated view that is always current

These are just a few examples of the types of applications that can benefit from CEP technology. The common denominator among them is that they share the need to continuously collect, process, and analyze data in real time, producing results without delay, even when the data arrives at very high rates.

While traditional databases were designed to process individual transactions at very high rates, analyzing the data to look for specific conditions or deriving higher-level summary data were tasks that had to happen offline, using query tools that were never designed to produce actionable intelligence in real time. This made them unsuitable for applications that had to analyze data in real time (such as trading applications). And while they were adopted for data analysis in other areas, the fact that the analysis is done on “historic” data means that the business insight is derived after the fact. In many cases, an opportunity is missed to react quickly to the results of the analysis.

Complex event processing technology delivers the data analysis tools traditionally provided by relational databases or even spreadsheets. But CEP delivers those tools in a real-time, event-driven implementation capable of processing incoming data at very high rates and producing results with near-zero latency.

Think of it as taking some of the fundamental concepts of a relational database and turning them upside down. A traditional relational database is designed to collect data and store it, where you can then analyze it to filter the data, combine it, group it, search for patterns, derive high-level summary data, and so on. The analysis happens offline, not in response to incoming events. In contrast, an event processor (the heart of

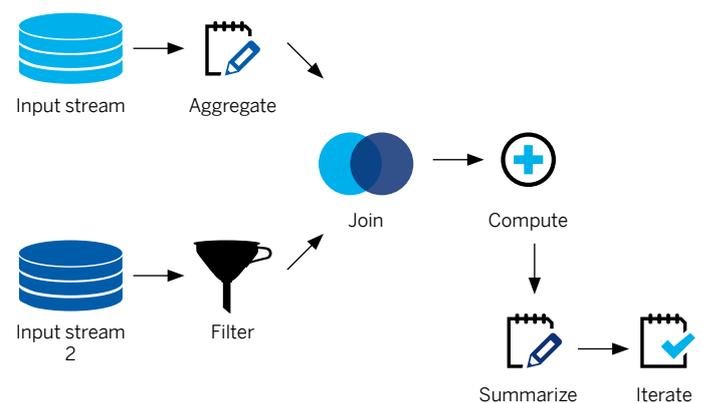
CEP technology) takes incoming messages and runs them through a set of predefined continuous queries to produce derived streams or sets of data.

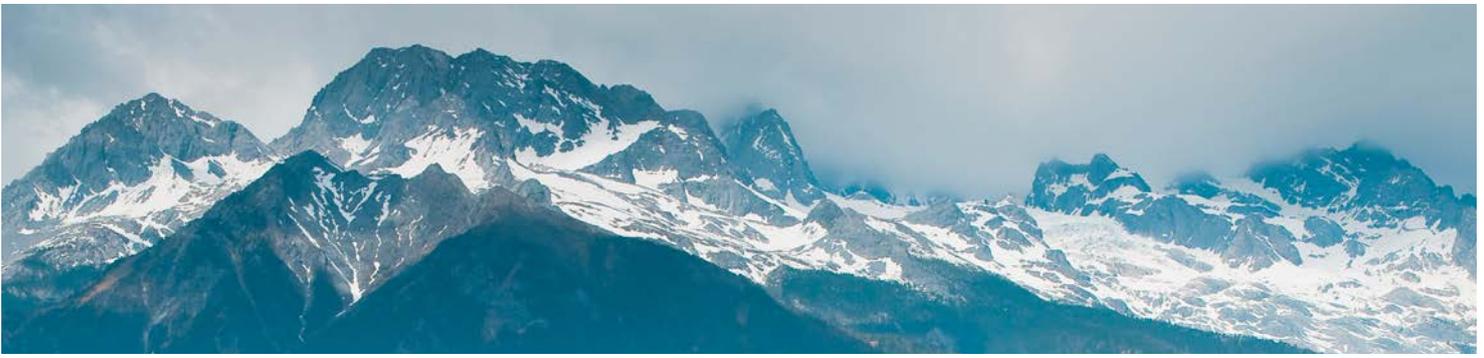
We call them continuous queries because the data analysis logic is similar to what might be included in a traditional database query. For example:

- Show me which events meet this criteria
- Tell me if this pattern of events (or nonevents) occurs
- Show me the current total of all events matching this criteria
- Group events by these values and calculate the average for each group

While complex event processing logic may be similar to a traditional database query, the implementation is anything but. Event processing uses a data-flow architecture to pass incoming messages through the continuous query operators as soon as the message arrives so that the result sets are instantly updated (see Figure 1). These functions used within the continuous queries have been implemented in a way to maximize throughput and minimize latency.

Figure 1: The Data-Flow Architecture





AN ALTERNATIVE TO CUSTOM CODE

SAP ESP provides an alternative approach to building high-performance enterprise-class applications that have to process event data in real time. Custom applications written in C++ or Java are expensive and time consuming to build. They are typically inflexible and therefore expensive to maintain since the processing logic is hard coded and tightly bound to the data structures. In fact, users have reported that, on average, building their applications with CEP technology from SAP reduced their development effort by 80% – as compared to building the application from scratch in Java. That not only reduces cost, but it reduces the time to get new applications into production, thereby improving business agility. It also means there's less code to maintain. By writing only the continuous query logic, and relying on SAP ESP for all the data management, you're maintaining only about 10% of the amount of code – leaving SAP to maintain the rest.

What's more, designing and writing highly efficient code for real-time processing requires specialized programming skills. This may not be significant for simple applications that don't need to scale or deliver high performance, security, high availability, and so on. But when you factor those things in, it's no longer just a simple application programming exercise. It requires multithreaded programming skills, an understanding of how to design code to interact efficiently with the hardware for maximum performance, and network programming for highly efficient and reliable I/O. By building on SAP ESP, all of those aspects are "under the hood," putting you in the driver's seat and letting you simply focus on implementing your business logic.

A NEW CONCEPT AND NEW TERMS

What often happens with the emergence of a new technology is that it takes a while for the industry to adopt common terminology and agree on common definitions. This is certainly the case with complex event processing. Over the past few years as this technology has emerged, it has been referred to as "event stream processing," "stream processing," or simply "event processing." In this paper we talk about "complex event processing" since that has become the label most industry participants are using to refer to this class of technology. But you'll notice that the SAP product for CEP is called an "event stream processor." Event stream processing is a form of complex event processing, and we believe it's the best description of what SAP ESP is designed to do.

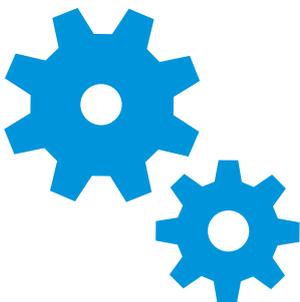
A SPECTRUM OF USES AND REQUIREMENTS

Regardless of the specific terms used, all event processing applications set out to do one or more of the following:

- **Situation detection:** Monitor incoming events to detect patterns that indicate the existence of an opportunity or a problem – a situation that warrants a response or that needs to be recorded. This can range from a simple filter to a complex set of rules that correlate incoming events and screen for sets of conditions that may include missing events. High-level events indicating the existence of the situation are generated as the result.
- **Data aggregation and analysis – continuous computation:** Data is correlated, grouped, and aggregated, and computations are then applied to produce new information such as summary data, high-level statistics, or adjustments to key operating parameters. This is often delivered to end users via "dashboards," or this summary information can be used by downstream applications that act on it. Examples of this type of CEP include:
 - Continuously adjusting prices based on movements in the market or other real-time inputs
 - Continuously updated key performance indicators (KPIs)
 - Continuously updated valuations and exposures
 - Continuous aggregation of data from multiple sources to see the big picture

- **Data collection:** A by-product of CEP is often the collection of raw event data or higher-level summary data. The collected data can be used as context for processing newly arriving events and can also be stored in a historical database for offline analysis, reporting, or an audit trail.
- **Application integration and intelligent event handling:** Many applications are built on an event-driven architecture, but the basic tools for this architecture provide the mechanisms for the exchange of event data without providing the ability to analyze it. CEP can provide intelligence within an event-driven architecture to analyze events in the context of other events. It can also provide insight into the state of various systems to determine what new events need to be generated or to determine the action to be taken based on an event. You might think of this as a streaming alternative to traditional extract, transform, and load (ETL) – but expand that to include an alternative to ETL that is not only real-time but has an additional intelligence layer. This additional layer can not only transform events but combine and filter them to produce new events based on business rules.

This is an important point, since in a particular context you may find the focus to be on one particular aspect of event processing. Yet recognizing that different applications have different needs will help you select the optimal tool or tools for the job.



SAP ESP is a [high-performance, enterprise-class](#), complex event processing engine that can be used to quickly implement and deploy a wide range of applications that need to analyze and act on event data in real time.

REAL-TIME DATA ANALYSIS

When we talk about the ability to analyze incoming event data in real time, we are actually referring to a variety of functions that can be applied to the data, alone or in combination, to derive high-level intelligence or to trigger a response. Examples include:

- Filter data to apply simple or complex filters to detect conditions of interest. This can include correlation of events across multiple sources, correlation of events across time, and watching for sets of events that match a defined pattern.
- Combine data from multiple sources, including the ability to combine streaming and static data or to combine data that arrives at different times. Define data retention “windows,” either based on time or number of elements, across which the computations will be performed.
- Group and aggregate data, producing high-level summary data and statistics. This can include trends (moving averages), net positions and exposures, and so on.
- Compute new data elements: enrich simple event data by adding new fields that are computed based on context, data from other sources, and so on.
- Transform data format and structure. This can go beyond simple message-level transformation and can create entirely new events based on individual or multiple events using rules that take into account context, reference data, and so on.
- Generate high-level events from patterns or groupings of low-level events.

SAP Event Stream Processor

SAP ESP is a high-performance, enterprise-class, complex event processing engine that can be used to quickly implement and deploy a wide range of applications that need to analyze and act on event data in real time. This product represents the state of the art in complex event processing, combining performance, versatility, and ease of use in an enterprise-class implementation designed for use in the most demanding environments.

DESIGNED FOR PERFORMANCE

The primary design objective of SAP ESP was, from the beginning, maximum throughput with minimal latency. On a 2-CPU Linux server, for example, SAP ESP can process well over 100,000 messages per second, with latency, measured from the time a message arrives until processing is complete and results have been produced. Results are typically in the range from a fraction of a millisecond to a few milliseconds.

Designed to be highly scalable and to function as an infrastructure component in mission-critical applications, SAP ESP is a multithreaded 64-bit engine supported on Linux, Solaris, and Microsoft Windows. The SAP ESP engine is fully multithreaded to take advantage of the parallel processing capabilities of multiple CPU machines, and it has clustering features allowing applications to scale across multiple servers. The net effect is virtually unlimited scalability, as additional CPUs and servers can be added as needed to increase throughput.

DESIGNED FOR VERSATILITY

Because different applications have different needs, many CEP products are designed to address a single type of application. For example, there are a number of CEP “rules engines” that are designed expressly for situation detection. That’s fine if all you need is situation detection, but the technology may not be extensible to other types of applications that rely more on continuous analysis of sets of event data. SAP ESP was designed to address the widest possible range of event processing requirements:

- Monitor incoming data streams for conditions that represent opportunities or threats
- Augment data streams with data from other sources or computed values
- Group data by different dimensions, producing high-level summary data or statistics
- Consolidate data from multiple heterogeneous systems, forming a single aggregate view or stream
- Operate on large data sets spanning large time windows
- Collect raw or result data for use in historical analysis, reporting, or to provide an audit trail

Some of the specific aspects of the SAP ESP architecture that give it this versatility include:

- Simple, declarative Structured Query Language (SQL)–style query language combined with a procedural scripting language for extensibility and control – The basic event-processing language used by SAP ESP is continuous computation language (CCL) – an adaptation of SQL for event processing. This leverages the familiarity and simplicity of SQL for the most common relational operations. However, there are often types of event processing logic that aren’t easily expressed in a declarative SQL statement. For those, SAP ESP provides a built-in scripting language called SPLASH that can be used to extend CCL with custom operators and functions.
- State management – Incoming messages can be processed as inserts, updates, deletes, or upserts. This lets SAP ESP address situations where incoming messages don’t just represent a new data point in a time series but represent an update to previous information. Most CEP implementations don’t handle updates and deletes; they treat all incoming messages as new data points in a time series. The reality is that many data streams produce updates, changes, and cancellations. Whether that involves changes to an order book or a correction to data previously sent, these updates need to be applied to previously received data to maintain an accurate view of the current state.

- Private cloud cluster architecture – The advanced cluster architecture of the SAP ESP engine enables it to function as a private cloud providing CEP services for any number of applications running on any number of servers where applications can be started and stopped dynamically. What's more, the cluster manager provides namespace management and URL resolution, allowing for efficient interproject communication within the cloud.
- An on-demand query interface that allows all event windows to be queried by external applications as if they were held in a database – The Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) interfaces provide support for off-the-shelf query applications and allow for snapshots of current data sets.
- Flexible subscription options – These options allow users and applications to customize their output streams by subscribing with SQL predicates, and when subscribing to windows, the option of receiving current state followed by updates.

DESIGNED FOR PRODUCTIVITY

Any CEP product is, after all, a productivity tool. You could build it all yourself in C++ or any other general purpose programming language, but building your application on SAP ESP saves you time and effort and results in a more robust, maintainable application. So it's important that the application development tools maximize user productivity.

With SAP ESP, there are two elements geared toward maximizing user productivity.

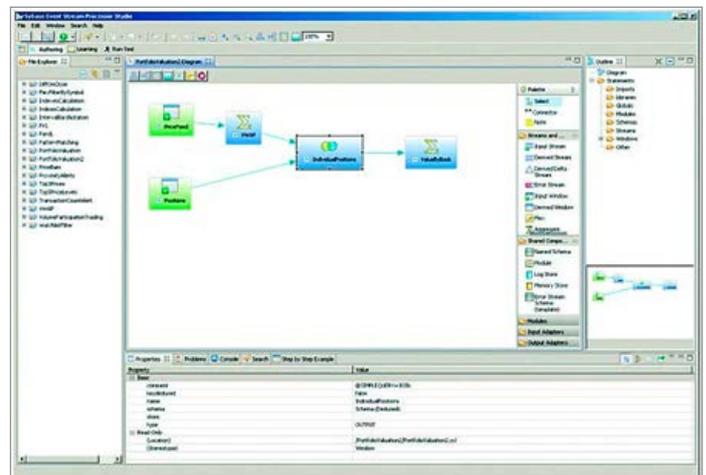
CCL + SPLASH: CCL is the event processing language supported by SAP ESP and is the language used to express the business logic for processing and analyzing incoming events. SPLASH is a simple scripting language that provides extensibility to CCL, allowing the user to quickly and easily write custom operators and functions. This powerful combination is designed to provide the richness and versatility required while maintaining simplicity to maximize user productivity.

The SAP ESP studio: This Microsoft Eclipse–based interactive development environment (IDE) provides all the tools needed to easily and efficiently build, test, tune, and deploy SAP ESP projects (see Figure 2). A unique aspect of the SAP ESP studio is that the user can work with projects in either a visual data-flow editor or a textual CCL editor. Power users will enjoy the efficiency of the CCL editor as well as the ability to visually see the data flow between operators, which is particularly useful for large, complex projects. New users and less technical users will appreciate being able to work entirely in the visual editor without having to learn CCL syntax. Users can switch between the visual editor and the CCL editor at any time with a simple keystroke, and changes made in one editor will be reflected in the other.

What's more, the SAP ESP studio includes a rich set of tools for testing, debugging, and tuning applications. The run/test perspective includes:

- Record and playback: capture live event streams and play them back at controlled rates for regression testing or back testing of new models
- Data upload from files and manual data entry tool
- Event tracer: see how each event affects the state of all nodes in the data-flow graph
- Debugger: set break points
- Viewer: see the output of individual streams and watch the contents of individual windows

Figure 2: Editor in the SAP® ESP Studio



SYSTEM INTEGRATION

SAP ESP receives events from systems that produce them, publishes output events to consuming applications, and makes SAP ESP windows available to query via ODBC and JDBC.

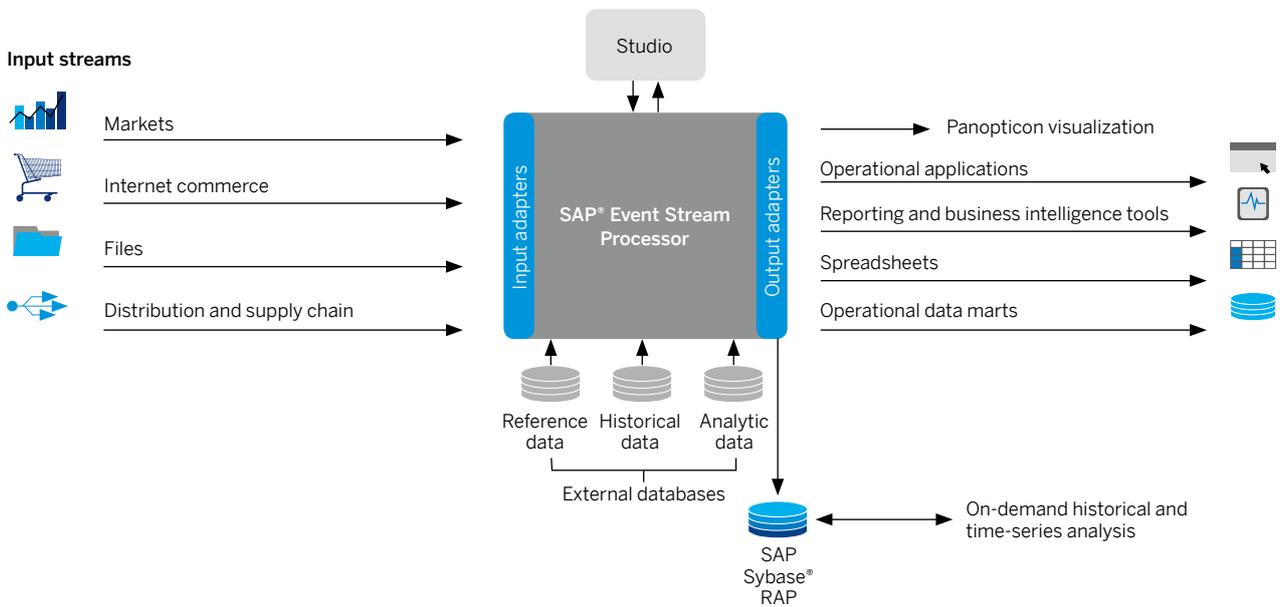
Typically, events are streamed into and out of an SAP ESP project via adapters (see Figure 3). SAP provides a wide range of adapters for connecting to common systems including:

- Message buses
- Databases
- Files and sockets
- E-mail (Simple Mail Transfer Protocol [SMTP])
- Web services
- Financial market data
- Financial information exchange (FIX)

In addition, the SAP ESP software development kit (SDK) includes interfaces and tools for building custom adapters or for enabling applications to integrate with SAP ESP directly. The SAP ESP SDK is available for C and C++, Java, and Microsoft .NET.

The on-demand query interface allows snapshot SQL queries to be run against any of the event windows within an SAP ESP project. Both ODBC and JDBC interfaces are supported. Command and control of the SAP ESP engine is via an XML remote procedure call (RPC) interface.

Figure 3: Events Streaming into and out of an SAP ESP Project via Adapters



SAP EVENT STREAM PROCESSOR PRODUCT FEATURES

The following is a brief summary of key features of SAP ESP.

Event windows: Event windows can be defined on a time or size basis. Older events are automatically removed from the window according to the retention rules defined. Windows allow for aggregation of events and computing aggregate statistics on sets of related events, and they also allow for lookups and joins.

Automatic state management: While most event processors treat all incoming data as a time series, appending the latest message to the history of the stream, the SAP processor uses a more sophisticated model that can also apply incoming messages to event windows as updates or deletes, automatically and efficiently maintaining the current state of dynamic data sets.

Data capture: Raw or derived data can be captured in SAP IQ software or SAP Sybase RAP software, providing a historical record for on-demand historical analysis or audit purposes.

On-demand queries: In addition to producing streaming output, all SAP ESP event windows can be queried via standard ODBC and JDBC interfaces. To a reporting tool, it looks like a database where the views are always current.

“Private cloud” cluster architecture: The SAP ESP engine can run as a cluster, effectively providing a private CEP cloud, running any number of SAP ESP projects across any number of machines. Projects can be started and stopped dynamically and can interconnect with other projects, with the SAP ESP cluster manager providing URL resolution as well as namespace isolation.

SPLASH scripting: Custom operators and functions can be quickly and easily written using the built-in SPLASH scripting language. SPLASH provides extensibility to CCL, significantly enhancing versatility without sacrificing user productivity. SPLASH is a small language designed expressly for working with events, tightly integrated with the streams and windows in CCL, and fully integrated within the SAP ESP studio.

External function libraries: SAP ESP has an interface for external function libraries, allowing proprietary functions written in C, C++, or Java to be invoked from within an SAP ESP query.



Building your application on SAP ESP **saves you time and effort** and results in a more robust, maintainable application. So it's important that the application development tools maximize user productivity.

Asynchronous and synchronous message options: While most CEP implementations assume full asynchronous operation, SAP recognizes that some applications need to ensure data integrity. Thus, synchronous message transfer options are available for input streams with tunable parameters to balance the needs for data integrity and performance.

Subscription options: Subscription options include the ability to subscribe to windows with the choice of receiving the initial window state followed by updates, and also the ability to subscribe with an SQL statement to customize the subscription with filtering or aggregation.

Built-in security: Security is supported by stream-level access control, authentication, and optional encryption of input and output streams.

Cluster configurations: SAP ESP supports clustered hardware configurations, including the ability to run a single data model across multiple machines with a cluster manager that will manage the distributed process.

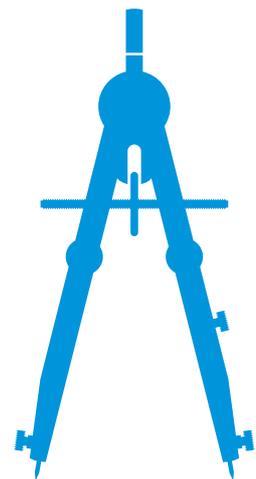
High-availability options: Hot standby facilities provide for automatic failover from a primary to a live secondary. Alternatively, a cold-spare configuration can be used to automatically start a new server on available hardware. Disk-based data recovery, which can be enabled for individual windows or all windows within a project, allows for recovery of window contents when a project is restarted.

SAP ESP studio: The studio is designed to maximize user productivity. It offers both visual and textual editing with the ability to switch between the two at any time, plus a full suite of testing, debugging, and tuning tools.

Adapters: A rich set of adapters enabling easy integration with common systems that produce event data to be processed by SAP ESP, hold static data that will be used by SAP ESP, and consume the output of SAP ESP.

SDK: The SDK for C and C++, Java, and Microsoft .NET enables development of custom adapters and integration with custom applications.

In the ever-changing financial services arena, speed, data volume, and accuracy are paramount for [maximizing profits](#), minimizing risk, and maintaining compliance with corporate and governmental regulatory requirements.



Typical Complex Event Processing Applications

SAP ESP enables the rapid development of a wide range of event processing applications. Just a few of the applications that have been built with this technology include the following.

Market insight: Filter, combine, cleanse, and analyze live market data. Select or combine data from multiple sources. Monitor data latency and quality. Compute value-added fields. Consolidate prices or order books across multiple venues. Compute market statistics. Analyze market depth.

Automated or algorithmic trading: Process high-volume market data to discover and act on trading opportunities; execute large orders using a variety of trading algorithms that track the market; consolidate depth-of-book data from multiple markets, enabling analysis of full market depth and pressure. In conjunction with SAP Sybase RAP providing tick history, algorithms have access to historical trading patterns – and new algorithms can be back tested.

Market making and autopricing: Cleanse and validate incoming market data and then apply skew and spread, incorporating trader input to update internal or published rates.

Pretrade validation and compliance: Check incoming orders for errors and compliance without adding latency.

Trade monitoring: Continuously monitor and report on trading activity, whether automated trading algorithms or human traders. Identify underperformance and trading outside of predetermined parameters, and track progress in filling large orders.

Best execution and smart order routing: Route orders in compliance with Regulation National Market System (Reg NMS) and Markets in Financial Instruments Directive (MiFID), taking into account protected orders, customer profile, and preferences and venue characteristics. Capture and record data for compliance monitoring and reporting.

Posttrade monitoring and reporting: Detect discrepancies, track performance, generate quality statistics, and generate alerts when acceptable thresholds are exceeded.

Risk aggregation and real-time profit and loss: Aggregate position and risk information, in real time, from multiple independent trading systems. Position-keeping systems and risk management systems provide a real-time aggregate view that can be analyzed across multiple dimensions, and nonintrusive consolidation of information in real time across organizational “silos” without the need to change or replace the underlying systems. Apply real-time market data for continuous valuation.

Data orchestration: Intelligently manage data dissemination within a service-oriented architecture, defining complex rules that incorporate knowledge of context and state to keep distributed systems in sync.

TAKE ADVANTAGE OF NEXT-GENERATION TECHNOLOGY

Critical business processes and decisions increasingly rely on having the most up-to-date information possible and the means to instantly react and respond to changing conditions. This is particularly key in the ever-changing financial services arena where speed, data volume, and accuracy are paramount for maximizing profits, minimizing risk, and maintaining compliance with corporate and governmental regulatory requirements. SAP Event Stream Processor, a high-performance complex event processing platform, offers next-generation technology for meeting these demanding data challenges now and into the future.

For more information, please visit our Web site at www.sap.com/solutions/technology/database/complex-event-processing.



CMP22160 (14/01)

© 2014 SAP AG or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG (or an SAP affiliate company) in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices. Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP AG or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP AG or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP AG or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP AG or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP AG's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP AG or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.



The Best-Run Businesses Run SAP™