CoreCard Software White Paper Series

CoreENGINE Software Platform

A Powerful Way to Build, Deploy, Operate
And Change Financial Service Applications

October 2009

CoreCARD SOFTWARE

Table Of Contents

Introduction

Financial organizations face unprecedented competitive pressures to introduce new products and services quickly, while simultaneously improving customer service and reducing costs. New government regulations enacted in response to recent financial industry turmoil require program modifications as well as more transparent customer service practices and notifications. Mergers, bank closures, and increased portfolio sales emphasize the importance of managing operational risks and responding rapidly to an ever-changing marketplace.

But change doesn't stop there. As technology developments lead us to Web 2.0 (and beyond) and more seamless integration of applications, many organizations find themselves redesigning fundamental business processes and attempting to cope with entirely new ways of conducting business. With new business models and entities emerging and prospering, older methods of doing business – and the technologies that automate those methods – are increasingly becoming a hindrance to competitiveness and obsolete.

A major challenge facing financial institutions is the need to react quickly to changes in the market. Unfortunately, this fast rate of change is often not reflected in the way mission-critical information and transaction management systems are constructed and work.

Many financial organizations are built upon a technology base that is a patchwork of incompatible technologies all feeding into and out of COBOL behemoths. Others are built on massive ERP solutions that were originally designed to support a manufacturing environment. Both of these platform types make responding to change a complex, time-consuming and costly process.

No doubt about it – a major barrier to reacting quickly often stems from a lack of flexibility in software technology.

Technical Challenges for Financial Services

Keeping Up With Current Needs

The financial services industry faces a number of significant technical challenges in the 21st century business environment. Information technology (IT) innovations are emerging at and multiplying with increasing rapidity, but the industry has built-in handicaps that hinder it from using most of these innovations to its best advantage. For example, the information and transaction management systems so critical to financial services performance have been built over time on a technology base that is complex, inflexible and costly in terms of both equipment and people. Typically, these systems have been designed, built and maintained as stand-alone units. Take the example of a company that has developed one application for billing, another for authorizations, and still another for collections. All too frequently, no attempt is made to exploit the commonality among these applications, and exchanges of data are usually done via batch updates. These applications are programmed to compute with hard-coded business logic, and many expensive professional staff members are needed to develop, maintain and modify system software running on large mainframes.

Problems Faced When Using ERP Systems

Even in organizations that have implemented large scale ERP solutions, many of these ERP applications are a patchwork of independent systems that have been developed to operate independently of each other. Making the related components talk to each other is a major undertaking, creating an industry for integration consultants to build and maintain these complex systems. In the end, these solutions still don't deal with the redundant code built into each which can create additional problems.

This situation has resulted in information and transaction management systems that are quite inflexible and expensive. This is a particularly pressing problem for smaller organizations seeking to implement new services, without limited capital or professional technical resources.

The Challenge of the Internet

One particularly important demand is to facilitate e-commerce, customer care and account management over the Internet. Today companies need to respond quickly to new business needs and especially to web-based customer demands. Unfortunately, financial services information and transaction processing systems are often not integrated, are difficult to modify and are expensive to maintain and change. Often systems require another separate business application or set of applications be layered onto the legacy system in order to meet new business needs – or they have to be replaced. Evidence is the array of applications interfacing to legacy applications on mainframes throughout the industry.

A Unified Application Solution

CoreCard Software, Inc., believes there is a better way and has created a suite of products built to use CoreENGINE® its universal transaction processing platform. .

CoreENGINE is a network-centric, highly flexible, real-time software platform designed to solve the hard problems inherent in financial service applications. While CoreENGINE itself is not an application in the typical sense of the word, it provides the foundation needed for a financial service or billing application with powerful, pre-built components.

Typical financial service applications are built to address specific processes or functions. The result is that organizations often have separate systems for new accounts, authorizations, case management and other elements of financial transaction account management. Not only does this approach mean having to modify several separate applications should a change be required, it is time-consuming to develop new applications that meet new and evolving business processes. This is why, over the last several decades, organizations have continually tweaked and tinkered with existing code, attempting to update applications instead of creating new applications.

CoreCard recognized that this never-ending cycle required a different approach to the development of financial service applications. Instead of going in for a "quick fix" that would only create future problems, CoreCard took a step back from the mindset that financial applications must be developed for specific purposes. Instead, CoreCard asked a fundamental question that is at the heart of why CoreENGINE is so different from existing technology platforms: ***What is common across all these applications?***

This simple question allowed CoreCard to view application development with a fresh perspective. All financial service applications boil down to three basic elements:

- Accounts
- Transactions that are posted to those accounts
- Rules by which those transactions are processed

Accounts, transactions, and rules: the basic elements and building blocks for any financial service application. The key difference from one application to another is how the data is defined: what type of account, what type of transactions, and what type of rules.

While most component-based programming techniques focus on defining the methods and inheriting those methods across the development environment, CoreENGINE gets to the heart of the matter. The challenge was to build a platform that would allow the implementation of what is common across all financial applications in a consistent way, while also allowing what is different to be implemented in a way that was unique and appropriate for that particular application.

CoreENGINE Benefits

CoreENGINE brings substantial benefits to applications with the following characteristics:

- Those centered on customers that have financial accounts.
- Financial transactions include but aren't limited to:
  - Loans
  - Purchases
  - Payments
  - Fees
  - Interest
  - Points
  - Services Charges
  - Rebates
  - Authorizations
  - Chargebacks, Presentments, Re-presentments, etc.
- Hierarchies are used for controlling the operation of accounts and maintaining balances.
- The accounts and hierarchies track balances of various kinds such as:
  - Principle
  - Interest
  - Services Charges
  - Loyalty Points
- Transactions are reflected in the balances according to a set of rules.
- There is a periodic rendering of a statement or invoice.
- There is systematic and controlled interaction with customers via a call center or web-based customer self-service.

CoreENGINE Business Goals

CoreCard designed CoreENGINE to be a solution that would enable financial services applications to be built, deployed, operated and changed with lower cost and greater speed than by any other means. In short, CoreENGINE was created to solve difficult, real-world financial service information and transaction system problems.

The architects of CoreENGINE had three main goals for defining and evolving the technology:

1. Acknowledge the need for flexibility and develop a solution that is built for change.
2. Provide technology and business innovations which allow businesses to compete by reducing cost.
3. Support online business applications by providing a framework for truly Internet-centric applications.

CoreENGINE Design Principles

The design of CoreENGINE represents a fundamental shift in the way financial service applications are defined and built. The software platform and the applications built on it t are based on a combination of well understood advanced technologies and a carefully selected group of techniques.

There are five significant CoreENGINE design principles:

- **Objects and Components:**

  The system is object-oriented in terms of construction method and application design. The base objects define common capabilities of the system and application objects define particular user requirements. Both kinds of objects are gathered and deployed as components to maximize re-use and interchangeability.

- **Message-Based:**

  The system is based entirely on messages. This architecture separates the processing of work from the source and destination of data. This enables systems to be configured to match any architectural requirement without changes to application definition.

- **Repository-Based Information:**

  The system is intentional and declarative rather than procedural, with all component attributes stored in an active repository that fully uses inheritance and polymorphism. The repository enables changes that affect many aspects of an application to be declared once, thus streamlining the process of customization.

- **Workflow Engine:**

  The system is centered on a workflow engine that provides rules-based branching, routing, priorities and work assignment among all processing units and people involved in the system.

- **Data Flow Architecture:**

  The CoreENGINE platform builds applications designed to run in a network of workstations (NOW architecture). This architecture may be used to achieve any combination of data throughput and response time via machine availability. This also allows scalability to meet increasing workloads, and delivers economy of processing power due to the use of lower cost machines

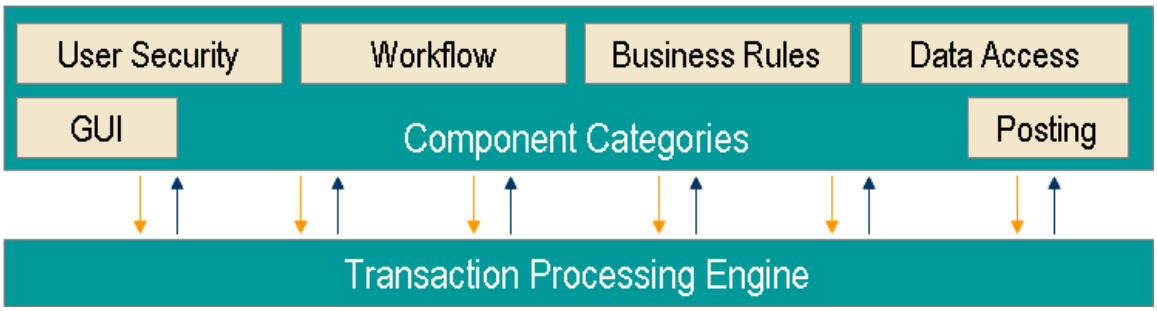
Patented Technology

CoreCard is building next-generation systems on an entirely new technology base, resulting in some advanced architecture features. CoreCard has one patent issued and one pending patent application:

1. Methods and systems for managing financial accounts that avoid batch processing and provide parameter based approaches (issued)
2. Methods and systems for performing workflow (pending)

CoreENGINE Technical Architecture

The CoreENGINE architecture is designed for complete separation between user interaction code, core processing code and DBMS. While most client/server applications are built today according to a two-tier architecture (fat client, DBMS-only server, possibly with stored procedures), most experts see that scalable and enterprise applications require a three-tier architecture (medium client, applications server, database server with stored procedures).



The CoreCard software platform is built in two layers. The lower layer, the transaction engine, consists of a set of highly generalized C++ classes and a program/administrator's interface to take advantage of available class libraries and programs. The upper layer consists of objects, data tables and user functions that adapt the behavior of the classes to the customer's situation.

The CoreENGINE Transaction Processing Engine provides the basic behavior of the system. Among the objects and capabilities it defines are messages, queues, rules, events, look-up tables, summing operations, interest/fee calculators, editors and constraints. A wide variety of entities more familiar to the user are built and/or defined in terms of these generic objects, including accounts, hierarchies, transactions and balances.

CoreENGINE Independence

CoreENGINE is built to be hardware and environment independent. Because of this, it can work with current technologies as well as those to come in the future.

- **Platform Independence:**

  The application is independent of the machine and operating system it runs on. The few necessary system dependencies are encapsulated and controlled by the administrator, not the application programmer.

- **Data Source Independence:**

  Applications are written to be completely independent of the method of storing and retrieving the data. The method to be used for a data set is controlled by the system administrator and can be changed without impacting the application. Data stores include in-memory lists, files, DBMS tables, network connections, IPC, queues, host files and programs.

- **SQL Independence:**

  SQL is the data access language used by RDBMS's. Although it is standardized, industry vendors use somewhat varying versions, and the standard changes periodically. Furthermore, SQL may not be used to access other data sources. SQL violates data source independence, and is therefore not part of the application. CoreENGINE generates dynamic and stored procedure SQL as required to meet the application's data manipulation requirements.

- **Configuration Independence:**

  Applications are defined in the same way for single-processor use as they are for LAN/WAN networks of hundreds of clustered computers. To support large user counts or high availability, it is usually required to build multi-tier (minimum three) applications. The tiering is normally explicit and part of the application definition itself, sometimes by means of explicit calls to a TP monitor. In systems powered by CoreENGINE, the application definition is absolutely independent of the configuration on which it runs. The system administrator controls the number of processors, machines, networks and connections on which the application runs and can make changes at any time.

- **Network Independence:**

  Applications are defined independent of the networks involved in their operation. When a unit of work in CoreENGINE gets something to do, it never knows where it comes from. When CoreENGINE generates results or sends out new work, it never knows where it goes. The application programmer determines the logical flow of work among worksteps. The system administrator controls whether the work goes through database tables, TCP/IP connections, IPC queues, or any other supported method.

- **GUI Independence:**

  Applications powered by CoreENGINE are independent of the user interface on which they run, while at the same time they make use of all "native" features of supported interfaces. Long lasting commercial applications cannot be tied to a particular interface, but must also conform to the latest standards by being native. CoreENGINE supports these conflicting requirements. All CoreENGINE-enabled applications support Windows® and web interfaces with no additional application work required.

- **Universal Integratability:**

  CoreENGINE-enabled applications are open in the sense that they provide an API and support calling and being called from external applications written in any language. Because of the message passing and open workflow architecture, it is possible to get work from and give work to CoreENGINE-enabled applications essentially without limitations. In addition, it is possible to "attach" to external data stores and network connections and adopt those external standards.

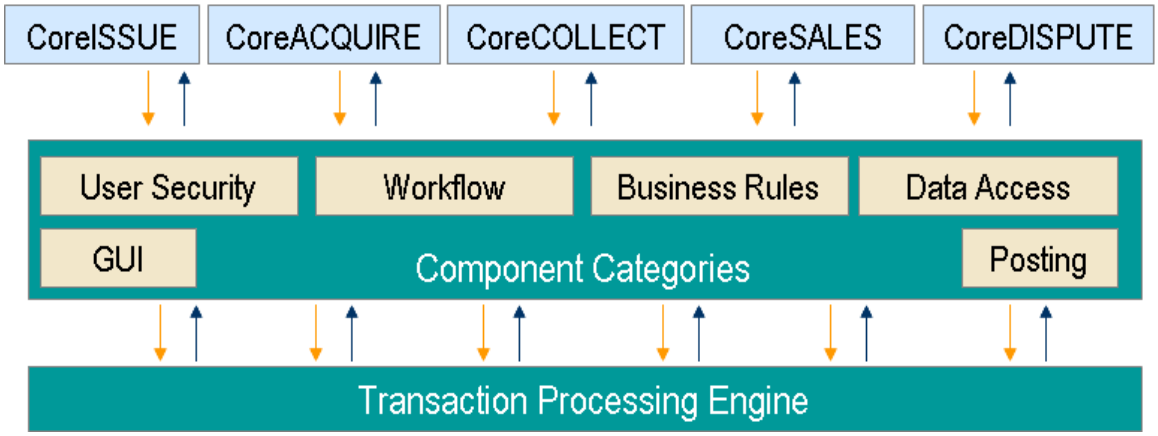
- **Language Independence:**

  National language support is an integral part of CoreENGINE. All applications have inherent support for all languages and dialects. Language is based on the CoreENGINE principles of inheritance, so specific dialects can inherit from a parent language (e.g., Spanish as the parent, Puerto Rican Spanish as a child). Applications will auto detect the language of any browser and default to the requested language or the closest parent language.

CoreENGINE-Enabled Applications

CoreCard built into CoreENGINE its accumulated knowledge about accounts, rules and transactions to resolve the hard problems ubiquitous to financial service applications. The result is the ability to define an account in most any manner that an organization needs, and to change account structures and hierarchies quickly.

With CoreENGINE, CoreCard has moved the technology concepts of abstraction and reuse up several layers, closer to the businesses and the processes that need their support. Previously, programmers have used application components, but until now they have not been assembled into a comprehensive set of business components engineered to solve the general problems of a given domain – in this case, financial services.



Nor has the same type of flexibility enjoyed by programmers using C++ or Java been part of a software platform that does not require programming to build new applications. The CoreENGINE design departs from the old ideas of application development. Furthermore, CoreENGINE is focused on a specific set of problems defined in the form of pre-built components that are designed to meet the unique needs of financial service applications.

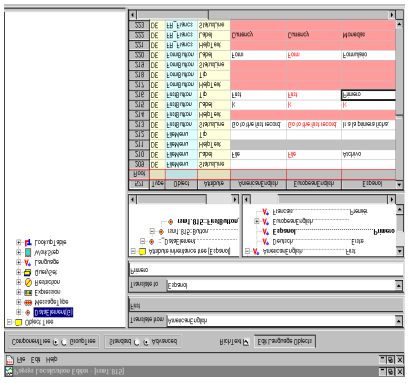
Some of the reusable components in CoreENGINE include accounts, account hierarchies, balances, transactions, aging, workflows, data warehousing, event tracking, messaging and security. These component elements are inherent in processing any type of transaction, managing accounts and enabling customer service or self-service via the Internet.

The modular components within CoreENGINE are inherent to the system, i.e., they do not have to be re-built as applications are developed. Flexibility exists in how an organization can define the components. Within CoreENGINE, each piece of knowledge is expressed exactly once. Consequently, by changing a component in one place, the changed component will "ripple out" everywhere in the system.

The result is a highly flexible, change enabling system. By using editing tools (e.g., Localization and Visual Workflow, which are used by CoreCard's engineers for end user customization), many aspects of a CoreENGINE-enabled application can be modified without writing a line of code and worrying about the impact of the change because CoreENGINE manages the changes for affected applications.

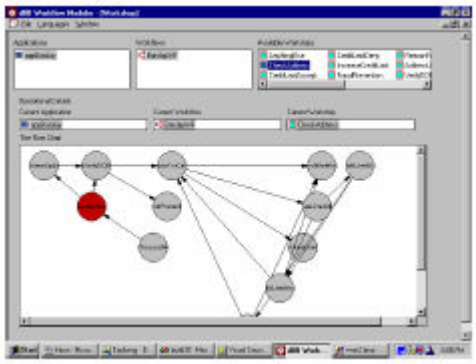
Localization Editor

An example of this change capability is the use of the multi-language support of CoreENGINE. By using the Localization Editor, an administrator can add a new language to an application and provide translations for labels, scripts, menus, etc. The change affects every aspect of the application, edit dialogs, help screens, tool tips, etc. Because CoreENGINE application screens are configured at run-time, the new language support is handled automatically. Fields are positioned appropriately based on their size and moved as needed based on the language in use at the time.



Visual Workflow Modeler

The Visual Workflow Modeler can be used by an administrator to visualize the specific work/logic path of a workflow and to easily prepare "models" for alternative or expanded work management. Workflow, routing rules, work assignments and much else involving both people and system processes can be examined and altered by supervisors using the editing feature, without otherwise changing an application.



Highlights of CoreENGINE and CoreENGINE-Enabled Applications

- Performs authorization and posting in real-time.
- Runs on multiple processors in a single box, and uses as many shared-nothing boxes as are available, providing linear scaling, fault tolerance and the ability to add capacity while the system is running.
- Supports native Windows® and HTML users from a single application definition for both customer service and self-service.

- Supports extensible user-defined hierarchies, with multiple parents, variable depth, etc.
- Multi-lingual, user-extensible text of all kinds (labels, titles, help, tool tips, etc.), including support for Asian languages.
- Built-in general workflow facility that provides for case creation, assignment, routing, rules-based branching and tracking.
- Built-in Wizard Workflow facility that provides scripting and just-in-time training for all user modules.
- Complete security control per user and/or user group. Advanced level of data security and permission granting with create, read, update and delete for ALL fields. Fields that are not allowed are invisible to the user.
- Comprehensive edit checks on field entry, with look-up tables, state tables, cross-field edits, multi-lingual error messages, etc.
- Standard RSM (result set manager) button set, providing for full featured searches, selection and refinement of results, intuitive navigation among results, insert, update, delete, and form/grid mode switch.
- Built-in navigation facilities that enable a user viewing a panel to navigate to related panels by clicking a button. Navigation works for walking a hierarchy (go to parent, go to children) as well as going to panels not related by hierarchy.
- Facilities for users to make changes to screen arrangement, security, text, etc., without affecting source code.
- Extensive built-in application integration methods, including transparent read and update access to both CICS and Tandem™.

Benefits of CoreENGINE

The CoreENGINE software platform is CoreCard's enabling technology for building, deploying, operating and changing financial service applications cheaper and faster. Applications built on the CoreENGINE software platform offer significant business benefits that generally fall into three categories: flexibility, Internet-centric, and lower costs.

- **Flexibility:**

  CoreENGINE designers aimed to improve flexibility in three key areas: hardware, software and people.

  In today's business environment, hardware changes rapidly. It is important for systems to be able to expand hardware capacity by adding the best available equipment, with those machines simply adding to the total processing capacity of the system.

  Software is basic to the daily functioning of today's business enterprise. Software that can be changed quickly, at a reasonable cost, and can be deployed quickly

with reliable quality, will provide a tangible competitive advantage. CoreCard's software is "designed for change" in that it expresses most application concepts in few locations so it is easy to find and make changes. The flexible software also contains effective dating, so every change is "as of" a date. This enables easy updating of software and accurate calculations that span time periods (e.g., calculations that span a period of interest rate changes will be done automatically using the correct rates, even if reversals or other transactions take place).

Reality is that people are not very flexible. It is very difficult to have people perform their jobs well while adapting to change at the same time. CoreENGINE designers realized that everyone who has customer contact must consistently be prepared and effective regardless of constant change in policies and practices. Don't understand how CoreENGINE does this – just points out that people are inflexible and that we know it – but what do we do about it?

- **Internet Capabilities:**

  Effective use of the Internet is nearly always an important business goal in today's markets. Many businesses can only tap into the power of the Internet by converting or interfacing with their current applications. However, it is much more desirable to have truly Internet-centric applications, such as those created with CoreENGINE.

  Conducting business over the Internet includes Internet-native interfaces, which display the text in the national language of the reader, real-time and 24x7 operation, and extreme scalability and fault tolerance.

  All applications powered by CoreENGINE are multi-lingual and multi-currency to promote global Internet use.

  CoreENGINE-enabled applications have a real-time posting engine with parameters to set priorities so that CoreENGINE can automatically post transactions and have results available right away. This suits the nature of Internet use around the world, 24 hours a day, 7 days a week.

  The inherent scalability provided by CoreENGINE's dataflow architecture and posting engine supports growth in transaction volume. By adding machines and changing certain configuration parameters, the system ensures that enough processors are allocated to the specific unit of work so that calculations or other business processes are performed immediately upon being requested.

  When operating mission-critical financial applications on a large scale, 24x7, any downtime is incredibly expensive. The indirect costs may be even more staggering. CoreENGINE includes a fault tolerance layer that performs functions similar to those performed by the operating systems of fault-tolerant operating systems, such as Guardian (Tandem).

  CoreENGINE makes use of multiple machines and multiple networks to spread the load when all elements are working and to recover from faults when something goes wrong by working around the faulty element.

- **Lowered Cost and Process Improvement:**

  The design of CoreENGINE was not focused only on delivering a "best of breed technology", but was aimed at reducing three types of costs:

  - Hardware
  - Software
  - Personnel

  Even today, in a time of dramatically reduced hardware costs, most high-end software runs only on hardware that costs more than 10 times as much in terms of cost per unit of processing power than commodity type systems.

  Software costs are driven by having to buy additional systems (middleware, glueware, DBMS, etc.). Most financial software systems are composed of products from varied vendors, which increase the purchase cost, the cost of integration, and the continuing cost of keeping multi-vendor combinations running.

  CoreENGINE-enabled applications run most financial applications without the need for other software systems (although customers may elect to keep certain existing applications running for a period of time, which may require an interface to CoreENGINE). CoreENGINE-enabled applications include functions normally supplied by other software packages, including authorizations, CRM, customer self-service, middleware, etc.

  Moreover, CoreENGINE-enabled applications are typically priced using a transaction-based model, so the up-front software costs are low, and grow only as the revenues of the business increase.

  Personnel costs often are the most significant concern, because people interact with customers and help the system to succeed. Productivity rises when people have the opportunity to get more done and to do it with higher quality and consistency.

  CoreENGINE-enabled applications can minimize people cost, while enhancing quality and consistency at the same time. Additionally, the same tools and techniques which are used to provide "just in time" training and consistent application of business rules within an organization's call center can be used to push simple servicing scenarios to the end customer.

  This is achieved by means of the wizard workflow model, in which the software performs all functions that can possibly be performed in software, leaving people to handle only the exceptions. All the contents of a customer service training course are expressed in the workflows, minimizing the training time for new representatives, and the degree of segmentation that brought to customer interactions is unlimited.

  Customer support experts can handle the small number of exception cases that drop out of the workflows. At the same time, they can add new branches and

cases to the workflows, giving a degree of sophistication of response to customer requests to the newest worker that previously would have taken months of training and experience to gain.

About CoreCard Products

CoreISSUE

CoreISSUE® is CoreCard's browser-based solution for managing credit accounts and transaction processing for the card issuing (including prepaid cards) side of the credit business. CoreISSUE provides intuitive information organization, easy navigation, and the ultimate flexibility in cardholder management. CoreISSUE provides cardholder management in a client-server environment, creating working screens that have a familiar look and feel to users accustomed to accessing the Internet.

CoreCOLLECT

CoreCOLLECT® is CoreCard's browser-based solution for managing and working delinquent credit accounts. CoreCOLLECT is a rules-driven collections system that offers customizable workflows to handle and route accounts that are delinquent to the appropriate personnel within an internal or external collections department. CoreCOLLECT's case management capabilities and workflows automatically monitor the payment schedules. CoreCOLLECT enables collectors to set an individual Promise to Pay, or a series of recurring PTPs, on an account.

CoreACQUIRE

CoreACQUIRE® is CoreCard's merchant processing application, supporting authorization processing from an acquirer's perspective. CoreACQUIRE provides the ability to receive authorization requests from a network switch (or gateway) and routes authorization requests back to a switch (or gateway). The application also provides the ability to verify the merchant as part of an authorization request, to identify and process an authorization request for a merchant, cardholder, or for merchants and cardholders.

CoreDASHBOARD

CoreDASHBOARD® offers Operations staff members the ability to monitor and control all the servers where CoreCard applications such as CoreISSUE are running. Dashboard can be used to monitor individual computers and groups of computers, establish alarms that are generated if certain conditions arise in a single computer or a group of computers, send notifications via email when certain conditions and alarms are generated, run live queries against the database, and more.

CoreSALES

CoreSALES® is CoreCard's application for acquirers and issuers who want to pay commissions to sales agents. Interface output transactions and files are received from CoreISSUE, CoreACQUIRE and CoreCOLLECT for posting to CoreSALES. Activity from sales agents, sales managers, and related account activity is tracked, settled and reported accordingly. Commissions are settled at both the sales representative and sales manager level based on many types of activity, such as cardholder and merchant monetary activity. A commission notification (much like a cardholder statement) is available on the Internet.

About CoreCard Software, Inc.

CoreCard Software, Inc. (www.corecard.com) licenses transaction processing and account management software and offers boutique processing services as well.

CoreCard's solutions provide easy to use parameter-driven controls, real-time transaction processing, built-in fault-tolerance, and a fully scalable architecture. CoreCard's software provides the market's most feature-rich platform for processing and managing accounts receivables and a full range of card products including prepaid/stored-value, fleet, credit, debit, commercial, government, healthcare and private-label cards.

CoreCard's server-based architecture provides the speed, flexibility and control to effectively manage electronic and card-based payment products. CoreCard's software is ideally suited for program managers to create and manage card programs, merchant acquiring and authorization.

As a custom processor, CoreCard processes card programs with a built-in option for the customer to license the software and become its own processor in the future.

Headquartered in Norcross, GA, CoreCard is a subsidiary of Intelligent

Systems Corporation [NYSE Alternext: INS].

For More Information – Call 770 564 8000

Notice