Advanced search

IBM home | Products & services | Support & downloads | My account

IBM developerWorks : IBM developer solutions : IBM developer solutions articles

developerWorks

Show me the money!

e-mail it!

Electronic payment processing for Web businesses

Indran Naick (indrann@us.ibm.com), e-business Architect, IBM
Reema Gupta (reema@us.ibm.com), e-business Architect, IBM

February 2002

This article describes several general methods of collecting money on the Web. It discusses online credit card transactions, other types of payment, and different payment protocols. The authors also include details about a payment processing system using the IBM Multi-Payment Framework and, specifically, WebSphere Payment Manager.

Contents:

Introduction
Payment processing choices
Credit card processing
Other types of payment
Payment protocols
IBM Multi-Payment Framework (MPF)
IBM WebSphere Payment Manager
Summary
Resources
About the authors
Rate this article

Related content:

Subscribe to the developerWorks newsletter
More dW IBM developer solutions resources

Introduction

An electronic payment system is any kind of network service that includes the exchange of money for goods or services. When large amounts of money are exchanged it is called a *macropayment* system, and small payments involve a *micropayment* system. The amount of money being exchanged is a significant consideration when designing the payment system and deciding on security aspects.

The roles of the players in an electronic payment system can be summarized as:

Payers and Payees

Make or receive payments (payers and payees can be individuals or organizations).

Banks or financial institutions

Hold accounts for payers and payees.

Third-party nonbanking financial institutions

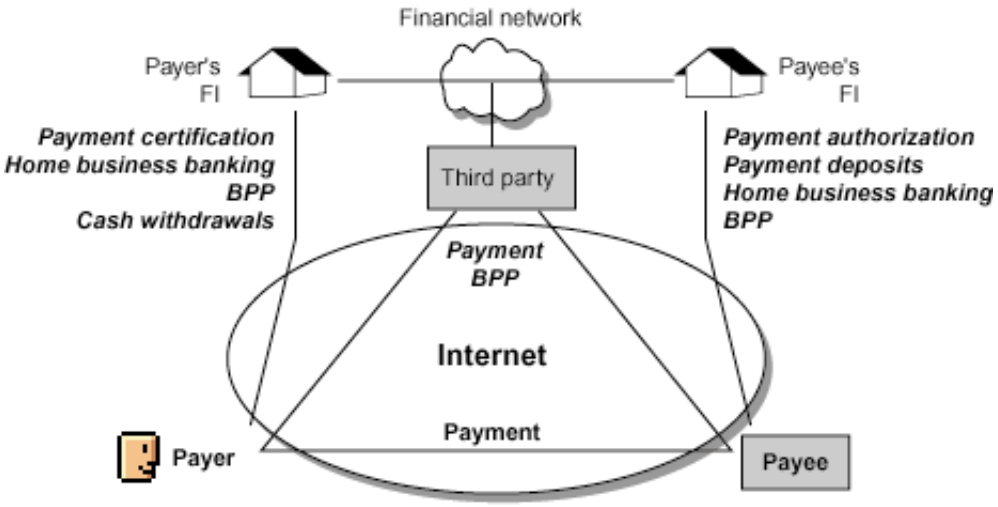
Provide payment services and interface with financial networks to activate transactions against accounts held in banks (CyberCash is an example of a third-party nonbanking institution).

Financial networks

Interconnect banks to each other and with third-party nonbank financial institutions. (MasterCard and Visa run credit-card networks designed for realtime payment authorization, whereas the Automated Clearing House (ACH) and wire transfer networks focus on batch clearing of transfers between accounts.)

Figure 1 below shows one scenario of participants in an electronic payment system.

Figure 1. Participants in an electronic payment system



Payment processing choices

This section describes some of the main types of payment processing options. Many other solutions can be customized for integration with these payment choices. Merchants select the most efficient way to handle payment processing primarily by considering the expected volume of online transactions, and their internal e-commerce capabilities.

- **Real-time payment on the Internet**

  Generally real-time payment processing over the Internet is provided with credit cards. To provide this type of payment processing, merchants must establish an electronic connection from their e-commerce site to an acquirer, which then connects to the card-processing network.

  Merchants who lack payment systems infrastructure for large transaction volumes and resources can outsource to a payment service provider. The service provider maintains connections to the credit card or bank networks, including real-time credit card authorization and batch settlement. It can also provide other services such as tax calculation and fraud detection. Selecting a service provider with connections to many processors gives merchants flexibility in choosing an acquiring bank.

  Some providers of outsourced Internet payment gateway services are CyberCash, CyberSource, iTransact, SkipJack, WorldPay, authorize.net, and firstecom.com.

- **Real-time, or batched, payment with in-house software solution**

  Merchants with inhouse e-commerce expertise and the extensive infrastructures required to support payment processing, and who are expecting high transaction volumes, can develop in-house transaction processing capability. It can be done either by building the connections with in-house resources or by purchasing payment-processing software, which might still require some in-house customization. This software allows an online retailer to plug the payment product into its e-commerce site and establish the required telecommunications links with the card or bank-processing network with which the company has chosen to work.

  A batch is a collection of payments and credits that are processed as a unit by an acquiring financial institution. Batches are associated with a given merchant and account. After some time period, or after a predetermined number of payment and credit transactions have been added to a batch, the batch must be settled in order for the financial institution to execute the batched transactions (to actually transfer funds from the cardholder's account to the merchants, or vice versa).

  Some providers of payment processing software are IBM, Trintech, HP Verifone, Atomic Software, Spyrus, and ICVerify and Tellan (both owned by CyberCash).

- **E-commerce hosted solutions**

  Merchants without in-house e-commerce expertise, and who are expecting low-volume transactions can, as part of a larger outsourced e-commerce workload, outsource payments to a full-service commerce service provider (CSP). CSPs integrate the front-end e-commerce function along with payment processing connections to the credit card or bank networks. There are many ISPs and ASPs

that provide merchant e-commerce and payment hosting solutions.

- **Redirection-based payment solutions**

  Some companies offer services that allow merchants to post data or link to their page for payment services. These companies then handle all the payment processing, and return a success/failure message to the customer. This type of service is relatively inexpensive. However, a drawback is that users might get a sense of leaving the Web site and could feel that the transaction is not secure.

Credit card processing
Processing online credit card transactions is the same as face-to-face credit card transactions, except that the online transactions have to be routed through the payment networks.

A bank that issues cards to its customers is a card-issuing bank, or issuer. It registers the card holder, issues a card and operates a card account to which payments can be charged. Merchants who want to accept payments must establish an Internet merchant account with a bank, called the acquiring bank, or acquirer. The acquirer processes the credit transaction through the card networks and then deposits the funds in the merchant's bank account.

To process credit cards in real time, merchants must establish a gateway connection from their Web site to the processing networks. Gateway connections can be made by purchasing and installing software solutions, developing in-house solutions, or by contracting with an outsourced gateway provider. Or, to avoid a gateway connection, merchants can capture the customer payment information and then re-key it into the in-store payment systems. Customer payment information can also be obtained by mail or telephone, called mail order/telephone order (MOTO) transactions.

The first step in online credit card processing is to get authorization by the issuer. The issuer verifies a cardholder's information by matching it with the information it has and returns the status. Authorization can be done in real-time using the gateway connection or in offline mode. In offline mode, merchants might use a batch-style processing architecture, whereby the credit cards are authorized in large batches at a scheduled time. In this case, customers won't be asked to wait for their card to be authorized, but may get a message from the Web site stating their order was received and will be processed. Later, the system might send the customer an e-mail with notification of a successful card authorization (or failure).

After authorization, merchants initiate the process to receive payment, which involves three steps: capture, clearance, and settlement. The payment information received from authorization is captured only after the purchased products have been shipped to the customer. This information is transmitted to the acquirer for clearance and settlement of the transaction. Clearance is when a credit card company collects data about a transaction from the acquirer and delivers the data to the issuer. The issuer will use the information to post the transaction to the cardholder's account.

Merchants electronically submit all credit card transaction data to the acquirer, who then credits the merchant's account for the total amount of the transactions and seeks settlement with the issuer. The acquirer sends the card payment instructions through the national credit card networks (Visa, for example). National credit card companies sort all of the credit card transaction data and transmit the data to the appropriate issuer.

The transaction is settled when the credit card company collects funds from the issuer and pays funds to the merchant's bank for the cleared transactions. The card issuer pays the national credit card company, using Fedwire as the payment channel, transferring funds by authorizing the Federal Reserve to electronically debit its account at the Reserve Bank for the net settlement amount and to transfer the funds to the national credit card company's settlement bank account. This bank then pays the acquirer, using Fedwire.

To summarize, the typical steps to process an electronic transaction involving credit cards are:

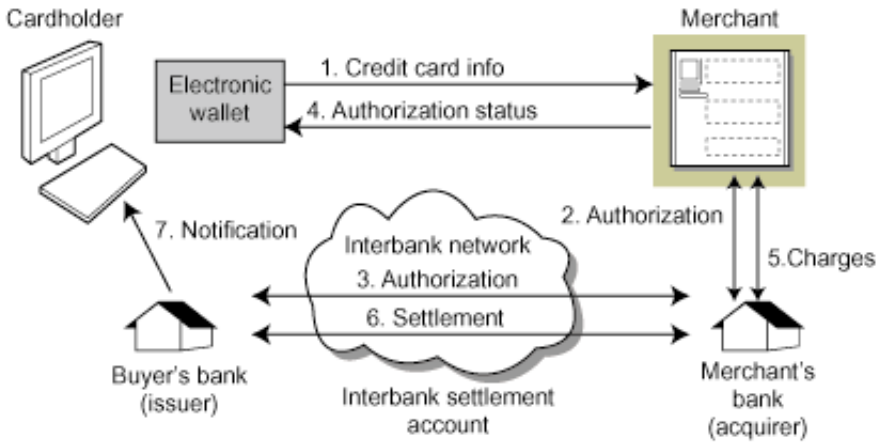
1. Buyer sends payment information to the merchant server. Merchant software must send and receive messages; encrypt and decrypt; store public and private keys; and request and receive certificates.
2. Merchant software takes payment information from the cardholder and sends it to the acquirer (merchant's bank). Acquirer institution must receive and authenticate payment information the merchant received from the cardholder.
3. Acquirer sends an authorization request to the issuer over the interbank network. Issuer sends the

authorization response to acquirer.

4. Acquirer notifies merchant about the status of authorization; if the response is positive, merchant fulfills the order.
5. Merchant presents the charge to the acquirer bank.
6. Acquirer sends a settlement request to the issuer.
7. Issuer charges the buyer's credit card account and at regular intervals notifies buyer of the transactions and accumulated charges.
8. Buyer pays the charges to the bank. Acquirer credits merchant's account.

Figure 2 below shows the flow of typical credit card processing.

Figure 2. Credit card processing



Other types of payment
The emergence of electronic shopping on the Internet has yielded new payment types. An electronic payment system can be:

- *Debit-based*, where the payer's account is debited immediately when the transaction is processed, or
- *Credit-based*, where the charges are posted to the payer's account and the payer later pays the accumulated amount to the payment service.

Electronic wallets, or e-wallets, are software programs that hold information about credit cards and other payment types (such as account number, expiration date, billing address, and so on) needed by buyers to do online purchasing.

Stored-value cards, which physically resemble credit cards but function differently, are used for low-value transactions when it isn't economical to use a credit or debit card. A stored-value card digitally stores an actual monetary value. When a transaction is made, the monetary value is instantly removed from the card. Stored-value card users do not need a bank account, and merchants need not verify the cardholder's identity when purchases are made.

Electronic cash, or e-cash, is a digital representation of a monetary value that is stored on the customer's hard drive. Customers uses this digital representation to pay for the transactions. Amounts of money can be withdrawn from a bank account and converted to e-cash. E-cash can be debited from a customer and credited to a merchant, or it can be paid to a merchant's bank account. The first company to produce an Internet-based electronic money product was DigiCash.

The electronic check, or e-check, system is basically an electronic implementation of the paper check system. The security features in e-check are encryption, digital signatures, and certificates. E-checks are widely used in business-to-business (B2B) electronic commerce transactions.

Telly-ho!

A few industry participants have their own payment methods. Telephone-based payments exploit the billing functions of telephone systems by allowing consumers of online goods and services to have payments charged to telephone bills.

Loyalty systems include a

B2B transactions use a different payment process, because the relationship between a business customer and supplier is long-term. The supplier accepts the risk of non-payment to encourage a long-term relationship with repeat orders. Suppliers are usually pre-qualified. Orders are placed as needed using a request for proposal (RFP) and bidding process. Electronic data interchange (EDI) and e-mail are often used to transmit orders because the customer-supplier relationship is established, and the ordering process is often driven by a corporate purchasing system rather than directed by a human being.

variety of merchandising incentive schemes, such as frequent flyer miles. Paypal can be used to send money to friends, e-commerce Web sites, Internet auctions or anyone with an e-mail address.

With B2B, products or services are usually delivered before payment. A supplier submits an invoice with the product, or bills later. Payment choices typically include paper check, e-check, ACH, Financial Electronic Data Interchange (FEDI), or wire transfer, all of which minimize costs but offer no protection against non-payment. These payment methods are asynchronous; customer and supplier don't need to meet face-to-face to complete the payments. Traditionally, payments are transferred by paper mail or banking networks such as ACH. On the Internet, the most common equivalent is e-mail. Business-oriented bill presentation and payment (BPP) systems are an alternative being promoted by some banks and third-party financial institutions.

Financial institutions interact with each other by Electronic Funds Transfer (EFT). For example, the interaction between the cardholder's issuing bank and the merchant's acquiring bank uses EFT. Three major EFT systems are the Society for Worldwide Financial Telecommunications (SWIFT), the Federal Reserve's Fedwire funds transfer service, and Clearing House Interbank Payment Systems (CHIPS).

Payment protocols

Payment protocols define the specific message format and information flow for transferring information between parties in an e-commerce transaction. The payment protocols should balance security, convenience, performance, and cost. The Internet offers two major modes of interaction: interactive (Web), and store-and-forward (e-mail). Credit and debit card payments are usually done interactively on the Web, and e-checks are used with store-and-forward e-mail.

The most commonly used payment protocols for secure online payment transactions are secure sockets layer (SSL) and secure electronic transaction (SET).

SSL

SSL is the standard for secure client-server communication on the Web. SSL protocol is a set of rules governing server authentication, client authentication, and encrypted communication between servers and clients. The two main advantages of SSL are simplicity and pervasive implementation in browsers and servers.

On the Internet, SSL can be used on the "front end" between consumer browser and merchant or at the "back end" between the merchant and gateway.

On the front end, SSL provides an encrypted transmission of Web pages, forms, and data between the browser and Web server. There are few limitations when used in this context. Merchants cannot later prove that a particular user authorized a transaction because SSL does not provide any provision for signing HTML forms. SSL secures the relationship between only two of the three participants (consumer, merchants, and banks), such as consumer-to-merchant or merchant-to-bank. And, SSL is limited in its use of certificates.

When used with the back end between merchant and gateway, SSL on its own does not ensure interoperability between the two. Merchants and gateways need a transaction protocol to define how they exchange data within the secure SSL tunnel. Usually these transaction protocols are proprietary, and specific to particular vendors and gateways.

The new Internet Engineering Task Force (IETF) standard called Transport Layer Security (TLS), which was published as an IETF Internet-Draft, is based on SSL.

SET

SET is an open technical standard for the commerce industry to enable secure payment processing over the Internet. It was developed by Visa and MasterCard with leading technology providers including GTE,

IBM, Microsoft, Netscape, SAIC, Terisa Systems and Verisign. The entire lifecycle of an electronic transaction includes the browsing, purchase, payment authorization, and payment capture processes. SET protocols define the last three of these. During a SET transaction, two things happen:

- *Authorization*, which is a check to make sure the buyer has credit or funds to pay for the merchandise, and
- *Authentication*, to ensure that both the buyer and merchant are really who they say they are.

SET protocol relies on public key encryption that uses financial institution, merchant, and cardholder digital certificates.

To ensure confidentiality of data, when the order information is sent from the consumer to the merchant the credit card number is encrypted separately from the order information. The merchant can read the order information, but not the credit card information. When the merchant requests authorization from the bank, the merchant sends the encrypted credit card information only; the bank does not get the order information, and only the bank can decrypt the credit card information. If the payment authorization is approved, the bank sends the required authorization number back to the merchant for their records. Only the information that is absolutely required by each party is actually sent to them.

SET has been extended to support merchant-initiated authorization (MIA). MIA permits merchants to accept credit card numbers by SSL-protected forms on the front end, while using SET on the back end between merchants and gateways. Thus, merchants can support cardholders who do not use wallets, while taking advantage of SET for interoperability with gateways provided by any vendor.

i-Key protocol (iKP)
iKP (*i* = 1, 2, or 3) is a suite of secure payment protocols developed at IBM Research Labs. They are based on public-key cryptography, but differ from each other depending on the number of parties that have individual public key-pairs and can thus generate digital signatures. The number of parties is reflected in the name of the protocols: 1KP, 2KP, and 3KP. As the parties who possess their own public key-pairs increase, the iKP protocols provides more security.

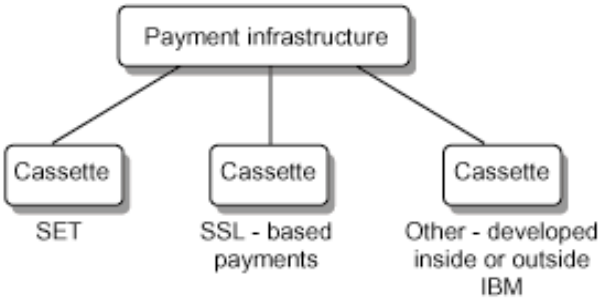
Protocol	**Requirements**
1KP	Only the acquirer needs a public key-pair. Buyers and merchants need only authentic copies of the acquirer's public key, reflected in a public key certificate. Involves a minimal public key infrastructure (PKI).
2KP	Merchants and acquirers need public key-pairs and public key certicates. Enables the customer and acquirer to verify the authenticity of the merchant.
3KP	All parties have their own public key-pairs and public key certicates, achieving nonrepudiation for all protocol exchanges. Payment orders are authenticated by a combination of credit card number, optional PIN, and digital signature of the buyer, making the forging of payment orders computationally infeasible. Enables merchants to authenticate buyers on-line. Requires a full public key infrastructure covering all parties involved.

IBM Multi-Payment Framework (MPF)
The IBM MPF is designed to:

- Provide a generic approach to multiple payment types and protocols
- Offer a consistent approach to user interfaces (UIs) and APIs across multiple payment types
- Provide a platform for implementing additional payment types and protocols.

A key element for achieving those goals is to use a plug-in technical architecture, which has a payment infrastructure and plug-in software cassettes for each payment type. The payment infrastructure provides the generic infrastructure functions required for making and receiving payments using any payment type, such as purchasing and bill payment in retail and business scenarios. Plug-in cassettes contain the implementation of specific payment methods and protocols. The IBM MPF supports multiple payment types, such as credit cards, e-checks, e-cash, and micro payments. Figure 3 below shows the MPF plug-in architecture.

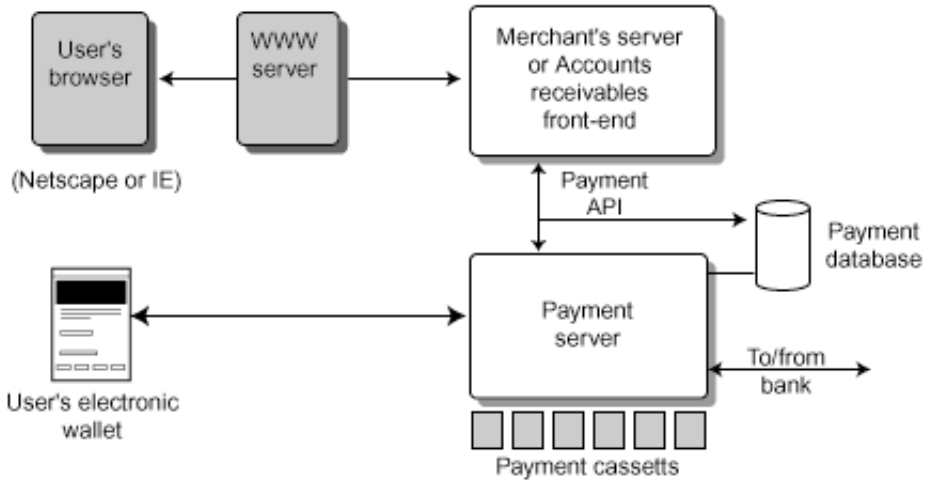
Figure 3. MPF plug-in architecture



IBM WebSphere Payment Manager
WebSphere Payment Manager provides a middleware "payment engine" for merchants. It can handle all payments, including tracking of payment status and communicating with merchant banks. When a user has a wallet, the Payment Manager handles the exchange of payment protocol messages with the wallet. When the user doesn't have a wallet, the merchant can request credit card data with an HTML form over SSL, then pass that data to the Websphere Payment Manager. In either case, the payment API allows the merchant server to control payment functions such as authorization, capture, and refund.

Figure 4 below shows how Payment Manager works with a merchant server or bill payment system to support payments on the Web.

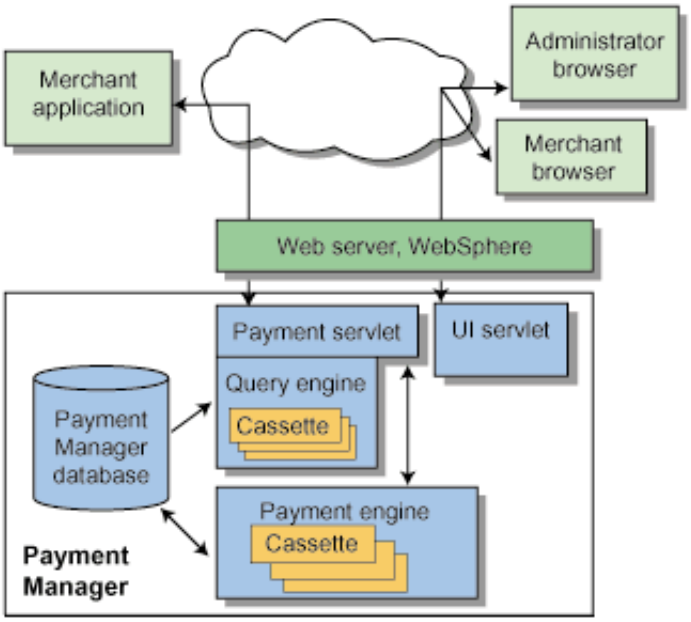
Figure 4. Payment Manager



Payment Manager is designed to run as an application under WebSphere Application Server (WAS). WAS provides the environment for Payment Manager to run its Java servlets, and the mechanism for external communications using HTTP requests (Payment Manager gives XML responses). The infrastructure uses IBM HTTP Server as its Web server and supports different relational databases, such as DB2, UDB, and Oracle. The Payment Manager uses a relational database to store Payment Manager configuration data, such as cassette configuration and authorization, and runtime data, such as transactional information for orders.

Figure 5 below shows the high-level architecture and main components of WebSphere Payment Manager.

Figure 5. WebSphere Payment Manager architecture



Payment Manager API
The Payment Manager API allows for integration with merchant applications. Access to Payment Manager is handled by a payment servlet. The Payment Manager API provides the merchant application with function calls for payment processing, queries and administration; an HTTP interface for low-level programming; a Java client API library (CAL); and a 128-bit SSL client.

Payment Manager UI
The Payment Manager UI permits interactive access to payment functions and parameters that are available using the API. The UI is used to make the Payment Manager (HTTP) requests. The XML responses to the requests are displayed using Payment Manager Presentation Language (PSPL).

The Payment Manager servlet runs under WAS and acts as the communication layer between the Web server and the Payment Engine. The Payment Manager servlet handles all API calls, interacts with Payment Engine, interacts with the payment database though the Payment Manager Query engine, and deals with all access control (access rights) whether through the UI or the API calls.

The UI Servlet handles GUI access. All requests passed to the Payment Manager servlet are by CAL methods. The UI servlet has one PSPL file with flags that point to a Java properties file. The Java properties file controls the look and feel of each screen. This lets you customize and brand the Payment Manager UI according to your needs.

Payment Engine
The Payment Engine is a Java-based application that gives the Payment Manager framework the flexibility to support different protocols. It deals with the payment requests that are sent to Payment Manager for processing. The Payment Engine handles the framework interface to the cassettes and the Payment Manager database tables.

The Payment Manager Query engine handles query API requests for viewing the database (for example, Query orders). Once the information is obtained by the engine, the response is formatted into an XML document.

Cassettes
WebSphere Payment Manager is a framework, and has extensions that plug into its framework. The extensions are pieces of software called *cassettes*. The framework manages all the common infrastructure needed by any electronic payment method, including access to the database, server redirects, definition of a language, and so on. The cassettes specify the flow of the electronic payment transaction, including the parties involved, the objects used, and their meaning. Without a cassette, the framework cannot do any electronic payment.

A cassette is the best way to create efficient code that manages payments. The core of the Payment Manager processes messages with a unique format. The plugins handle the translating and communicating

with other systems. Cassettes send and receive XML messages to and from the Payment Manager framework and translate them to the customer (usually merchant software) and to the acquirer (usually a bank or credit card holder).

Along with the framework, WebSphere Payment Manager installs two cassettes for testing purposes.

- The *OfflineCard* cassette enables the reading of card transaction details without being connected to a financial processing network. With the OfflineCard cassette you can view reports that consolidate both online and offline payments.
- The *CustomOffline* cassette enables the recording of arbitrary payment information, such as voucher payments, cash-on-delivery payments, or other payment schemes where money is not required to flow.

IBM has the following cassettes for WebSphere Payment Manager. There are also many third-party cassettes available.

Cassette	**Description**
SET	Provides support for 3D-SET, SET V1.0, and MIA, giving access to over 100 banks and processors in over 40 countries. Payment Manager uses the SET protocol to implement the generic payment processing model, commands, and objects defined by the Payment Manager framework. The SET protocol is request/response, where one entity (such as a cardholder) makes a request and gets a response from another entity (such as a merchant). The cardholder, merchant, and financial institution participate in the payment flows that make up a financial transaction. SET cassette supports order creation with wallet participation (using ReceivePayment command) and without. The MIA SET extension permits a merchant to use SET messages for authorization and capture orders placed by cardholders using a transmission method other than SET.
CyberCash	Allows users to access the CyberCash, Inc. CashRegister Credit Card Service through the Payment Manager programming and UIs. It admits payments with credit cards, electronic checks, and micro payments. To use the cassette for CyberCash, merchants must have a CyberCash-enabled merchant bank relationship in North America.
Visanet	Lets merchants send real-time Internet credit card transactions to VisaNet, the largest and most sophisticated consumer financial transaction processing system in the world, for processing. Provides worldwide telecommunications and payment data processing, authorizes and settles payments, and offers a range of value-added services such as risk management and fraud control. All VisaNet transactions are sent to the VirtualNet IP Gateway, a VirtualNet Internet Commerce Gateway, for processing. The VirtualNet IP Gateway provides connectivity via a private circuit. The private circuit is provided by the ISP, CSP, or merchant.
BankServACH	Provides connectivity to the ACH Network to support online electronic check payments.

Availability of cassettes changes frequently; for the latest information about cassettes from IBM and third-party vendors, check the WebSphere Payment Processing site listed in .

New cassettes can be developed to support new payment protocols. New payment methods can be added without having to modify and reinstall the core WebSphere Payment Manager application. Instead, a new cassette is simply added as a plugin to the base Payment Manager. To create new cassettes, IBM provides cassette developer toolkits that include a Cassette Developer Cookbook, Cassette Developer's Toolkit (CDT) and the PayGen test tool. CDT includes a programmer's guide, a sample cassette, and a set of utility programs to assist in developing and testing cassettes.

Summary

We hope you found this to be a useful discussion of general methods for implementing payment processing within commerce Web sites. Many payment types can be used, and several payment protocols are available. The IBM Multi-Payment Framework and WebSphere Payment Manager can greatly aid the integration of payment processing into your commerce Web site.

Resources

- See how the IBM dragonslayer team incorporated payment processing into their Go-ForIt application.
- Learn more about IBM Payment Processing from this WebSphere Commerce site.
- Access the Payment Manager documentation.
- Find all sorts of information on all the cassettes available for WebSphere Payment Manager.
- Download WebSphere Payment Manager cassettes (no charge, but registration is required).
- Explore the Payment Manager Cassette Developer Toolkits (no charge, but registration is required).
- Find more information on SET on the Secure Electronic Transaction LLC site.
- Read Netscape's SSL specification.
- Read the Transport Layer Security (TLS) IETF Internet-Draft.

About the authors


Indran Naick is an e-business Architect for IBM Developer Relations Technical Consulting in Austin, Texas, which provides education, enablement, and consulting to IBM business partners. Indran has over 14 years of industry experience. He joined IBM in South Africa in 1990. Prior to transferring to Austin, Texas he served as a Software Solutions Architect consulting to many financial and government institutions. He has authored a number of publications and is a graduate of the University of the Witwatersrand in South Africa. He can be reached at indrann@us.ibm.com.


Reema Gupta is an e-business Architect for IBM Developer Relations Technical Consulting in Austin, Texas, which provides education, enablement, and consulting to IBM business partners. She has held various e-business architecture and software development positions over her 7-year career with IBM and GE Capital Information Technology Solutions. She is an IBM Certified e-business Solution Designer, IBM Certified e-business Solution Technologist, and IBM Certified Specialist -- WebSphere Commerce Suite. You can contact Reema at reema@us.ibm.com.

e-mail it!

What do you think of this article?

Killer! (5) Good stuff (4) So-so; not bad (3) Needs work (2) Lame! (1)

Comments?

Submit feedback

About IBM | Privacy | Legal | Contact