Oracle® iPayment

Concepts and Procedures

Release 11*i*

January 2001

Part No. A86141-02

ORACLE®

Oracle iPayment Concepts and Procedures, Release 11*i*

Part No. A86141-02

Contents

Send Us Your Comments

Oracle iPayment Concepts and Procedures, Release 11*i*

Part No. A86141-02

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us via the postal service.

Oracle Corporation
CRM Content Development Manager
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Welcome to the Oracle iPayment, Release 11i. This *Concepts and Procedures Guide* provides information and instructions to help you work effectively with Oracle iPayment.

Intended Audience

This guide is aimed at the following users:

- Technical Service Representatives (TSR)
- Customer Service Representatives (CSR)
- System Administrators (SA), Database Administrators (DBA), and others with similar responsibility

This guide assumes you have the following pre-requisites:

- Understanding of the company business processes
- Knowledge of products and services as defined by your marketing policies
- Basic understanding of Oracle and Developer / 2000

Related Documents

For more information, see the following documentation:

- Apache Server Documentation (http://www.apache.com)

Conventions

The following typographic conventions are used in this manual:

Convention	Meaning
italic text	Book titles
`Courier text`	User commands and file content examples
UPPERCASE	Structured Query Language (SQL) commands, initialization parameters, profile options, responsibilities, or environment variables
boldface text	Menu, button, keyboard, and form options
< >	Angle brackets enclose user-supplied names. Note: Do not type the angle brackets.

1

Understanding iPayment

This topic group provides overviews of the application and its components, explanations of key concepts, features, and functions, as well as the application's relationships to other Oracle or third-party applications.

New in this Document

The following new features have been added to Oracle iPayment 11i.

Voice Authorization

The authorization API for credit cards and purchase cards has been expanded to include voice authorization. Voice authorizations occur when a merchant is required to contact a financial institution directly to perform an authorization, bypassing iPayment and the back end payment system (BEP). The financial institution gives the merchant an authorization code, which can be used to inform iPayment and the back end payment system that a voice authorization has occurred. Even though the actual authorization is performed outside iPayment, the voice authorization functionality allows merchants to perform follow up transactions through iPayment, similar to what the merchants would do after any other authorization. The back end payment system must support voice authorization for this feature to be used in iPayment.

Extensibility

Extensibility allows iPayment's interaction with a BEP servlet to be customized. Additional parameters may be added to the transaction immediately before it is sent to the BEP, and taken from the BEP's response and stored in the database immediately after the BEP servlet finishes processing the transaction.

This is done by implementing a Java interface provided by iPayment, which is loaded dynamically during execution of a payment request. If no such implementation is found, then the transaction proceeds normally. Otherwise, methods from the implementation class are called, which can then handle custom parameters in the BEP request/response."

CyberCash Servlet Performance Parameter

In a single merchant scenario, i.e. when the cybercash servlet only uses one Cybercash merchant, it is possible to improve the performance of the servlet by avoiding some overhead processing. Overhead processing is required when multiple merchants are used. A new servlet initialization argument has been added to the CyberCash servlet to enable or disable this performance improvement feature.

Currency Based Routing

By using the iPayment routing rules you are now allowed currency type as a rule condition. That is, you may now choose to send transactions to a certain payment system based on the type of currency it uses. This new rule condition exists in addition to other conditions of transaction amount and payment instrument type.

Overview of Oracle iPayment

Oracle iPayment provides an integrated electronic payment solution for both electronic commerce applications and client-server applications. It provides integrated, user friendly access, and control of payment processing to these applications. iPayment supports two electronic payment methods: credit card payments and bank account transfers. iPayment also supports payment partners such as Cybercash and CheckFree.

iPayment offers easy installation, administration, and extension capabilities. The risk management functionality of iPayment can quantify and identify the fraudulent online transactions for both business to business and business to consumer models.

Because there are few standards in electronic commerce and payment processing, iPayment supports several routing options, payment methods, processing models, and security features.

Key Benefits of iPayment

- Integrates with many payment processing systems. This feature allows businesses to offer several payment options to their customers and thus reduce implementation and maintenance costs.
- Provides rules based payment processing. This feature allows businesses to incorporate their existing business operations, rules, and procedures and lower costs by controlling relationships with payment processing vendors.
- Provides security through support for industry standards such as SSL and SET.
- Integrates with other Oracle applications, such as, Oracle iStore via Order Capture and Order Management, Oracle Receivables, and a single, open application programming interface (API) to integrate with any web-based, or client-server application.
- Provides support for both single and multi site installations of electronic commerce or client-server applications. iPayment also allows both stand-alone businesses and internet service providers to offer electronic payment processing.

iPayment's Integration with Other Oracle Applications

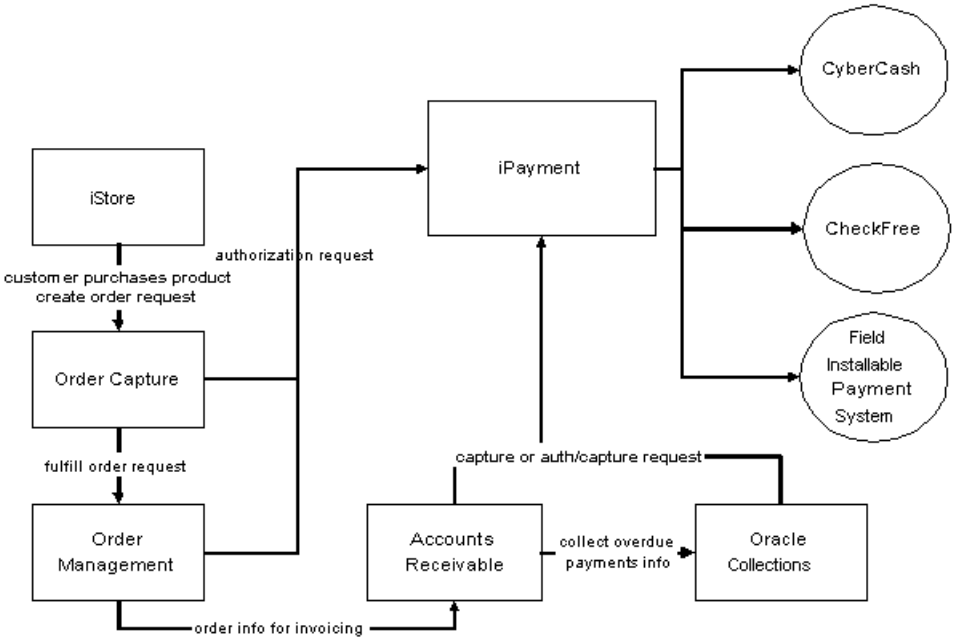
iPayment integrates with other Oracle applications to provide payment processing across your enterprise. Various applications send payment transaction requests to iPayment for processing. Without iPayment, each of these applications would need to build integration to the backend payment systems. iPayment saves integration effort by providing a single source to the backend payment systems such as CyberCash, Checkfree, and country-specific or region-specific payment systems.

Example of a Payment Processing Flow Using iPayment and Other Oracle Applications:

1. **Sales application (for example, iStore or TeleSales)**: Customer purchases product and decides to pay by credit card. The sales application submits the order.
2. **Order Capture or Order Management:** Order Capture and Order Management process the order. They use iPayment to verify if the credit card number is valid and authorize order amount. They may optionally perform some risk evaluation as part of the authorization.
3. **Accounts Receivables:** When the order is shipped, the credit card information is passed to Account Receivables and the billing and credit capture takes place.

4. **Oracle Collections:** When the payment is overdue and your call center begins outbound collection attempts, Oracle Collections uses iPayment to authorize and capture credit card transactions.

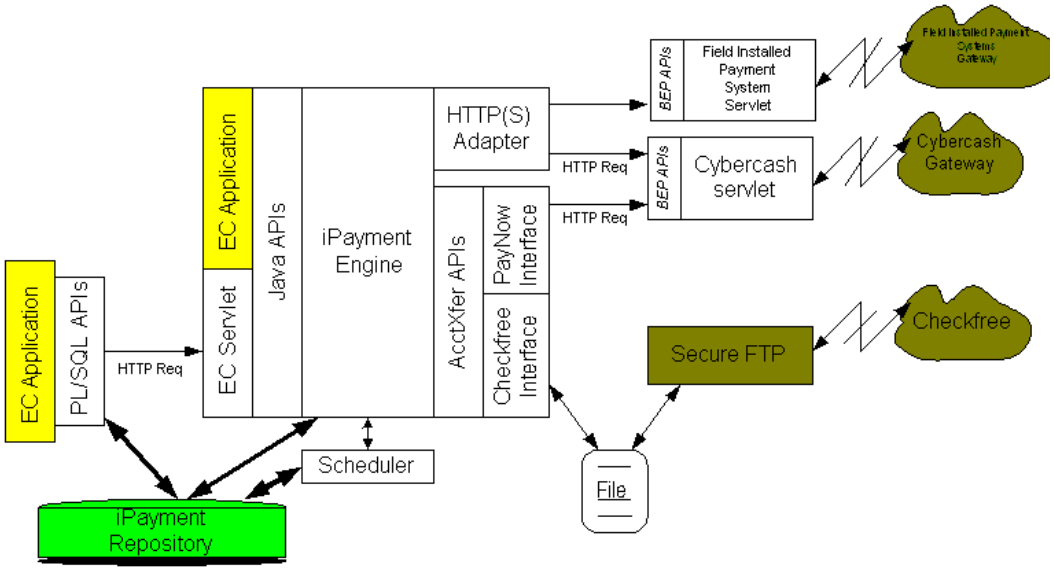
Figure 1–1 iPayment's Integration with Oracle Applications



iPayment Architectural Overview

iPayment can be integrated with any electronic commerce or other sales applications. The integrated iPayment component can communicate with Oracle database and other servlets to provide payment processing.

Figure 1–2 iPayment Architecture



iPayment APIs

iPayment provides two types of application programming interfaces (APIs) to simplify the integration of existing or new applications with iPayment for payment processing.

- **Electronic Commerce APIs**
  Electronic commerce applications can use these APIs to integrate their applications with iPayment. The electronic commerce application can be a servlet that plugs into any application server, or it can be a stand-alone application that communicates with iPayment via Java APIs or via PL/SQL APIs.
- **Payment System APIs**
  Developers can use these APIs to create payment system servlets. These servlets are usually interfaces that link the payment system software to iPayment to facilitate electronic payment processing.

iPayment Engine

The iPayment engine contains functionality for multi payment method support, routing, risk management, etc. It works easily with the APIs.

iPayment Servlets

iPayment consists of the following servlets:

- ECServlet

The ECServlet provides an interface to the iPayment engine to process payment related operations such as authorization, capture, return etc. This servlet is basically used for the PL/SQL APIs provided by iPayment.

- Payment system servlets

iPayment provides a complete payment solution, bundling payment system servlets developed by Oracle and its payment system partners. The payment systems communicate with the payment processors and the acquirers/banks to process payment transactions. iPayment includes payment system servlets for Cybercash.

- Field-installable servlets

iPayment supports field-installable servlets. These are payment system servlets not bundled with iPayment. This feature allows a payee to acquire a new, additional, or upgraded payment system servlet and configure it in the same way as the payment system servlets bundled with iPayment.

The ability to add field-installable servlets provides payment flexibility and allows new releases of iPayment and the payment systems to be independent of each other. It also enables electronic commerce applications to customize the payment system for their specific needs and regions.

Field-installable payment system servlets for iPayment are usually available from Oracle's payment system partners.

Understanding Credit Card Transactions

iPayment handles both credit card, purchase card, and bank account transfers. The following information explains the process flow for a typical credit card transaction.

Traditional Credit Card Transactions

Traditional credit card transaction processing involves a customer, a payee, an acquiring bank or processor, and an issuing bank.

A traditional credit card transaction consists of three phases: authorization, settlement, and reconciliation.

- Authorization

  The customer purchases goods or services and sends credit card information and payment instructions to the payee or business.

  The payee accepts the authorization request and sends it to the credit card processor through iPayment and the payment system.

  The processor matches the information with a database maintained by the card issuer (such as Visa or MasterCard) to determine if the customer has enough funds available to cover the transaction. If the funds are available, then the processor reserves the funds and sends back an authorization code.

- Settlement

  Settling transactions includes capturing authorized transactions, processing voids and returns, and batch administration.

  The payee issues capture, void, return, credit, and close-batch functions to the processor through iPayment and the payment system.

  The processor settles the payment with the issuing bank and causes the funds to transfer to the acquiring bank.

- Reconciliation

  Depending on the agreement between the payee and the acquiring bank, the acquiring bank sends daily, weekly, or monthly reports to the payee for reconciliation.

  The payee cross-checks transaction information in the database with the bank statement for reconciliation.

Voice Authorization

Sometimes credit card processing networks decline transactions with a referral message indicating that the merchant must call the cardholder's issuing bank to complete the transaction. The payment information in such cases is submitted over the phone. If the transaction is approved, the merchant is provided with an authorization code for the transaction. To facilitate follow-on transactions through iPayment for this voice authorization (for example, capture or void), iPayment provides voice authorization support.

Understanding Terminal-Based and Host-Based Merchants

The financial industry uses two processing models for credit card transactions: terminal-based merchant and host-based merchant. iPayment supports both. The following information describes these two processing methods.

- Terminal-Based Merchant

The payee or business determines when to close batches of transactions for clearing and settlement, and is responsible to perform close batch operation.The payee or business has more flexibility.

- Host-Based Merchant

In this model, the processor's host maintains all the transactions and is usually responsible for close-batch operation at a predetermined frequency. The payee or business does not have to perform close-batch operations. Corrections, such as returns and voids, are sent as new transactions to the host.

Why Is This Important?

The choice of being a terminal-based or host-based merchant is generally determined by the business type, number of transactions per day, and the model supported by the acquiring bank. The processing model you choose affects how you perform the settlement operations. For a terminal-based merchant model, you have to perform close batch operations periodically. Consult your acquirer for more information at the time of signing up.

Understanding Purchase Cards

Oracle iPayment supports purchase cards level II. Purchase cards are generally used in business-to-business scenarios. From an iPayment perspective, purchase cards are similar to credit cards, but more information is passed to the payment system using a purchase card.

Benefits of Purchase Cards

To the Merchant

- Accepting purchase cards is crucial to increasing competitiveness. Businesses use purchase cards to cut costs and streamline labor intensive processes to procure goods and services. Many buyers will prefer merchants that accept purchase cards.
- Merchants generally receive better rates for purchase cards than with credit cards.

To the Buyer

- A reconciliation stream by providing purchase order number and additional information.
- Aggregation of purchases when companies receive one invoice for multiple purchase cards.
- Streamlining the purchase order process. Lower processing costs by simplifying the purchasing process, reducing paperwork, and automating controls on the spending limits.
- Merchants accepting purchase card as a payment method help the buyer by making purchase information available electronically. This may help companies (buyers) comply with tax regulations, reporting requirements, and expense reconciliation.

Purchase cards are processed by most banks based on different levels of data which could potentially be sent from the point of sale.

Level I: The standard financial information existing on all credit card transactions for now. Example of Level I information includes:

- Supplier's name (same as merchant name)
- Total purchase amount
- Date

Level II: Level II information comprises of level 1 information and additional information to help the cardholder sort, reconcile, and report transactions. Example of Level II information includes:

- Purchase order number
- Sales tax amount

- Zip code to where the merchandise will be sent

Purchase card transactions are processed in the same phases as the credit card transactions. The phases are: authorization, settlement, and reconciliation. See Understanding Credit Card Transactions for more information about transaction phases.

The process flow from iPayment is similar for purchase cards and credit cards. But additional fields providing more information are passed from the electronic commerce application to iPayment, and from iPayment to the payment system.

The business flow differs on the buyer's side and for the payment system, but not for the merchant except for the additional information that is passed:

- Buyer places an order and provides payment information. Payment information is entered in the merchant's system. The information includes: purchase card account number, card expiration date, amount of purchase, applicable sales tax, and purchase order number.
- Buyer authorizes payment by requesting authorization through the payment system and the network.
- Card issuer verifies that the purchase is within the cardholder's authorized spending limits. Within seconds, the merchant receives either an approval of the payment request or a denial of the payment request.
- Merchant may display a receipt summarizing the items purchased, total amount of the sale, and any taxes paid.
- Merchant captures the payment by issuing capture to its processor.
- Funds are transferred from the issuing bank (customer's bank) to the acquiring bank (merchant's bank).
- Issuing bank bills and collects payment at the end of a billing cycle. The buyer receives a central invoice from the issuer bank for all company cardholder transactions.
- Buyer sends a consolidated payment to the purchase card issuer.
- Each cardholder also receives a monthly memo statement at the end of the billing cycle to review it for accuracy. This statement may be reconciled and approved by management.
- The buyer's accounting department allocates valid expenses to the appropriate project, cost center, general ledger, or purchase order account.

Understanding Bank Account Transfers

iPayment supports bank account transfers for both business-to-consumer and business-to-business models. Account transfer functionality facilitates electronic transfer of payment amounts from a customer's bank account to the payee's bank account using the ACH (Automated Clearing House) network. Electronic commerce applications use iPayment as their interface to payment processors that provide connectivity to the ACH networks. iPayment integrates with CheckFree and CyberCash's PayNow service to provide this new feature.

The number of operations supported in bank account transfers are fewer than those supported for credit card payments because of the current practices and processes involved in processing account transfers. You cannot receive real time response for bank account transfers due to the current practices in account transfer processing. The only status that can be provided is whether the payment was submitted to the processing network or not. iPayment only supports offline payments for bank account transfers. See Understanding Offline and Online Payments for more information.

Interface with Electronic Commerce Applications

Electronic commerce applications can use the same API for credit card, purchase card, and bank account payments. iPayment routes the request to the correct back-end payment system.

The operations that are supported for bank account transfers are merged into the same framework of operations that are supported for credit card payments. The following operations are supported for bank account transfers:

- Payment request
- Payment modification
- Payment cancellation
- Payment inquiry
- Payment query transaction status

> **Note:** Certain credit card operations are not supported for bank account transfers.

Process Flow of Bank Account Payment Request

1. The electronic commerce application calls the iPayment API to schedule an offline bank account transfer payment request.
2. All bank account transfer payments need some lead time before the settlement date. At the time of an API call, iPayment determines whether the payment request can be settled on the requested date or not, based on the lead time of the payment system.
3. If it can be settled, then iPayment accepts the payment request. Otherwise, based on the API parameters, iPayment either rejects the payment request or accepts the payment request with a different settlement date.
4. A scheduled offline payment request can either be modified or canceled before it is routed to the payment system.
5. Once a request is routed to the payment system, the electronic commerce application can neither modify nor cancel the request.
6. The payment system routes the payment to the ACH network.
7. If there is any failure on the ACH network or at the payment system site while processing the payment, then the payment system updates iPayment with those errors.
8. Finally, iPayment updates the electronic commerce application with all the status changes of the payment request.

Understanding Offline and Online Payments

iPayment supports two models of payment processing:

- Online payment processing
- Offline payment processing

Online Payment Processing

Online payment processing is the model in which payment processing request is immediately forwarded to the back-end payment processor. The results from the processor are immediately returned to the electronic commerce application.

Offline Payment Processing

Offline payment processing is the model in which payment requests are not immediately forwarded to back-end payment processors. When an electronic

commerce application makes a payment processing request in an offline mode, the payment information is saved in the iPayment database and is sent to the payment processor later.

The offline method uses a scheduler, a utility that functions at regular intervals. The scheduler browses the stored requests and sends requests to the back-end payment systems and updates to the electronic commerce applications.

Offline Bank Account Payment Request

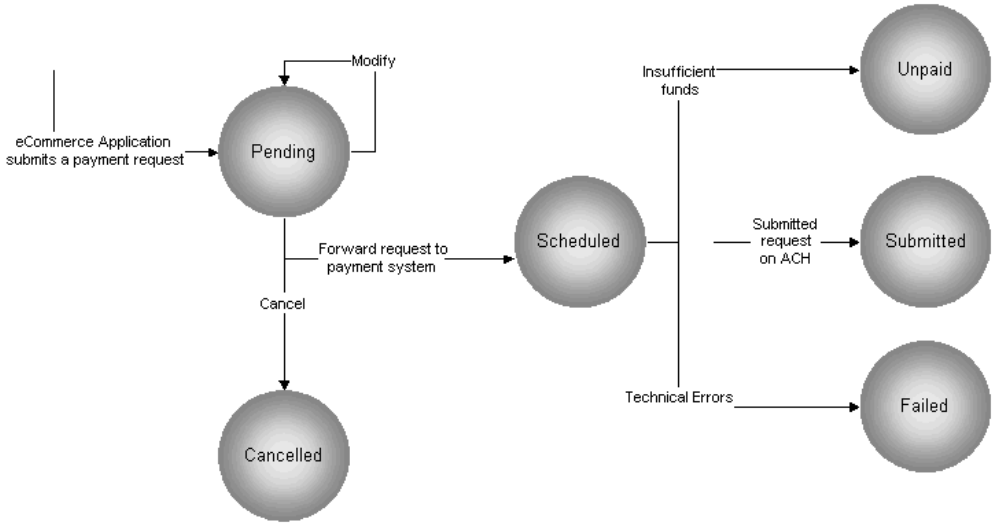
An offline bank account payment request in iPayment, at any given time, can be in one of the following states:

- **Pending**: After the electronic commerce application makes a request and before the scheduler routes the request to the payment system.
- **Scheduled:** After routing to the payment system.
- **Submitted:** Once the payment system submits the request to the banking network, for example, ACH network.
- **Canceled:** When a pending payment is canceled.
- **Failed**: Failed due to technical errors.
- **Unpaid:** Insufficient funds.

The state of a payment is determined on the status of the payment request. To obtain the status of a payment request, electronic commerce applications can call the Query Transaction Status API.

The following diagram shows the state diagram of an offline payment request (For bank accounts only).

Figure 1–3 State Transition Diagram of an Offline Payment Request-Bank Account

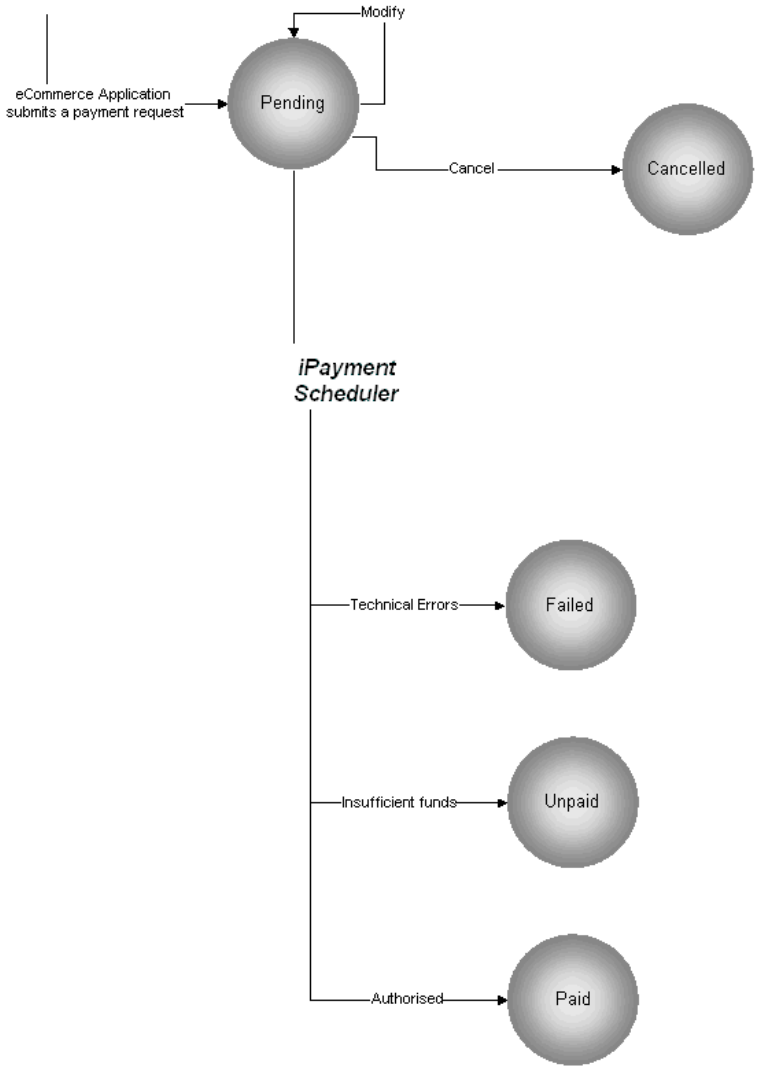


Offline Credit Card Payment Request

At any given time, an offline credit card request in iPayment, can be in one of the following states:

- **Pending**: After the electronic commerce application makes a request and before the scheduler routes the request to the payment system.
- **Canceled:** When a pending payment is canceled.
- **Failed**: Failed due to technical errors.
- **Unpaid:** Insufficient funds.
- **Paid**: Sufficient funds.

Figure 1–4 State Transition Diagram of an Offline Payment Request-Credit Card



How the Scheduling System Works

iPayment allows the electronic commerce applications to submit payment related requests for scheduling. iPayment's scheduling system periodically checks the database for pending payment processing requests and routes them to the appropriate payment system.

Scheduling System Flow

1. When iPayment receives a payment request, it determines which payment system should be used to process this payment request, based on routing rules.
2. iPayment stores the payment request in the database to schedule the payment at a later time.
3. At a configured time, the scheduling system scans the database for pending requests which have to be scheduled based on the settlement date. The scheduler also creates a list of the payments to be made and the payment system to be used for each request. The time interval can be configured in a cron job. The payments are scheduled depending on the requested dates.
4. The scheduling system sends the payment request to the appropriate back-end payment system.
5. The back-end payment system processes the payment request and sends the response containing the payment status back to the scheduler.
6. The scheduler updates the payment status in iPayment database and updates the electronic commerce application via the Updatestatus API which is implemented by the electronic commerce application.

Understanding Risk Management

iPayment provides risk management functionality for credit card and purchase card transactions for electronic commerce applications for both business-to-business and business-to-consumer models. iPayment includes a number of built-in risk factors and provides the option to the payees to run or not run the risk evaluation functionality for each payment operation. Payees can also run the risk evaluation for operations which handle amounts exceeding a specified amount.

A risk factor includes any information which a payee wants to use to evaluate the risk of the customer wanting to buy goods or services from the payee. Examples of risk factors are: address verification, time of purchase, payment amount, etc. These risk factors can be configured for each payee (merchant or biller).

Risk management functionality enables payees and electronic commerce service providers to manage the risk involved in processing transactions online. It allows businesses to have any number of predefined risk factors to verify the identity of their customers, assess their customer credit rating, and risk rating in a secure environment.

Payees can associate the risk factors with different weights as a formula and define any number of risk formulas in iPayment based on their business model. When a Payment Request API is called, the electronic commerce application can specify which formula to be evaluated to verify the identity of their customers, assess their customer credit rating, and risk rating in a secure environment. Alternatively, Risk API can be called independent of the Payment Request APIs. Using the Risk API separately allows merchants to first evaluate risk. Depending on the risk score, merchants may not want to send the payment request for authorization. Depending on the response from iPayment, electronic commerce application has to decide whether to process the payment request or to reject it.

Risk management helps businesses in reducing manual operational overheads to handle bad transactions and in avoiding costly penalties such as charge backs from banks.

Risk Factors Shipped with iPayment

The following is a list of basic risk factors shipped with the Risk management component. These risk factors can be configured per payee.

- **Payment amount limit** is the amount involved in the payment request. It varies from business to business and the risk factor score can be configured for different amount ranges based on the business model.
- **Time of purchase** is the time at which the payment request is made by the customer. Statistics reveal that most of the fraudulent purchases are made at midnight. Site administrators can define the time duration during which the payment requests are high risk and assign the risk factor scores for each duration.
- **Ship to/bill to address** is used to match the ship to address to the bill to address in the payment request. A payment request is considered high risk if these two addresses do not match.
- **Risky payment instruments** are a list of payment instruments (e.g, credit cards, bank accounts) that are considered risky by each payee. These include the instruments that were used by customers earlier and had resulted in fraud or chargebacks. Such a list can be generated internally by the payee or obtained

from other sources. If these instruments are reused in a payment request, then the payee may again face fraud or chargeback. Risk management functionality can detect if risky payment instruments are being used during processing by looking at the risky instrument repository. If the instrument being used for the payment is found in the repository, then the payment is considered a high risk payment. The Risky Instruments Upload Utility adds and deletes a list of risky instruments from the database.

- **Transaction amount** is the total amount of payments made by a customer using the same instrument in a specified duration of time. The duration of time is setup by the user. This is related to the payment amount limit risk factor. A customer can make payments in smaller amounts, which are not captured by the payment amount limit risk factor but can be captured by the transaction amount risk factor. Transaction amount risk factor sums up the total amount of payments in a specific duration of time and captures the risk on that amount. The transaction amount is an amount limit for a specified duration of time. The total number of payments made during a specific time period can be checked by looking at the payment history. The site administrator can set up a time duration and a transaction amount. In evaluating this risk factor, if the total payment amount exceeds the transaction amount within the specified time duration, then the payment is considered a high risk payment.
- **Payment history** tracks the reliability of the payor involved in a payment request. If a payor has a good history of payments over a long duration, then payments requested by this payor are considered to be low risk payments.
- **Address verification service (AVS) check** is the risk involved on the AVS code that is returned by the credit card network. Address verification service is provided by Mastercard and Visa credit card networks to match the billing address with the address that is maintained for the cardholder by the issuing bank. Various AVS codes are returned based on the complete address match, zipcode match, street address match, etc. A site administrator can configure all AVS codes returned by the payment systems and their corresponding risk factor scores. This service is only provided in US.
- **Frequency of purchase** tracks the sudden surge in the use of a payment instrument in a short duration. For a particular payment instrument in a payment request, if the frequency of use in a duration configured is more than the set up value, then the payment request is considered to be a high risk payment.

Oracle Receivables Risk Factors

For customers who have both iPayment and Oracle Receivables installed and registered, more risk factors are available. These risk factors are set up in iPayment and the values of these risk factors are setup in Oracle Receivables. Oracle Receivables stores credit management information about customer accounts such as credit rating, risk rating, etc. These are used in risk analysis.

- **Credit limit** is an overall credit limit associated with a customer's account. If a customer has an outstanding balance and the total amount of payment made by the customer exceeds the overall credit limit, then the payment becomes a high risk payment. Overall credit limit varies from business to business. It can be set up as an overall credit limit at the customer or site level through Oracle Receivables.
- **Transaction credit limit** is the credit limit per transaction associated with a customer's account. When a payment request exceeds the transaction credit limit, it becomes a risky payment. The transaction credit limit varies from business to business. It can be set up at the customer or site level through Oracle Receivables.
- **Credit rating** is the information that enables payees to effectively manage financial terms with their customers. It is useful for online financing or in evaluating purchases of a large amount by a new customer. It is a user defined field and the information can be taken from Oracle Receivables. A payee associates risk scores to credit rating. A higher risk score implies that selling goods / services to the customer is risky.
- **Risk code** is a user defined risk assessment field in Oracle Receivables. It is useful for online financing or for evaluating purchases of a large amount for a new customer. The information is available from Oracle Receivables. A payee associates risk scores to all the risk codes. A higher risk score implies that selling goods / services to the customer is risky.

iPayment Routing and Operation

iPayment accepts payment instructions from the electronic commerce application and routes them to the appropriate payment systems. The customer uses a web browser or a client application to exchange data with a web or server-based electronic commerce application. This application sends payment requests to iPayment. Finally, iPayment routes the payment requests to the appropriate payment systems.

How Routing Works

Routing of a payment transaction is based on a set of routing rules set up on the iPayment user interface by the iPayment administrator. A routing rule can have three conditions:

- One condition is based on the amount of the payment request.
- The other condition is based on the instrument type used in the payment request.
- The third condition is based on the currency type used in the payment request.

Routing rules are prioritized by an administrator. During processing, the rules are evaluated in the order in which they are prioritized.

If the values of a transaction satisfy both conditions of a rule, then the transaction is routed to the payment system associated with that rule. The rest of the rules are not evaluated for that particular payment request.

If none of the rules match the transaction values, then the transaction is routed to a default payment system, set up by the administrator, based on the instrument type used in the payment request.

If an electronic commerce application has to override the routing scheme, then it could process its own routing logic and send a routing rule name to iPayment. iPayment uses that routing rule name to find a payment system. In this case, a rule can be set up without any conditions and this rule is not used during evaluation.

The rule name should be the name of one of the routing rules which the administrator has set up. If the rule name is not the name of one of the routing rules that have been set up, then the transaction is routed to a default payment system.

iPayment supports credit card payments and bank account transfers. The payment methods available depends on the payment system that you decide to use.

Payees and businesses can customize how iPayment routes transactions to the payment systems using routing rules based on their business rules and the available payment methods. For example:

- A business sends all electronic payment transactions to a single payment system: Payment System A.
- A payee sends all small or micropayment transactions to Payment System A and all credit card transactions to Payment System B.
- A business sends all bank account transfers under $10 to Payment System A, and all other transactions to another payment system B.
- A business sends all transactions using US dollars to Payment System A and all transactions using other currencies to Payment System B.

Understanding iPayment Security

The following security features are recommended to guard against unauthorized access to data and iPayment services. In addition, Apache Server provides several types of authentication that you can use to secure servers, listeners, and servlets.

- Firewall Protection
- Secure Socket Layer
- Basic Authentication for Payment Systems
- IP Address Restriction
- Data Privacy

Firewall Protection

Oracle strongly recommends that you install iPayment and the payment system servlets on a machine inside the firewall.

Oracle also recommends that you use one of the following two configuration options to further reduce the risk of data being intercepted as it passes between different parts of iPayment:

- Install all the following components on the same machine:
  - iPayment
  - Payment system servlet
  - Electronic commerce application

- Or, use secure socket layer (SSL) to connect distributed components

Secure Socket Layer

If either iPayment (or its components) or the electronic commerce application is installed in a distributed environment, then Oracle recommends configuring SSL between iPayment and the payment system components.

Basic Authentication for Payment Systems

For setting up security for basic authentication, you have to perform some tasks both in iPayment administration user interface and in Apache Server administration tool. While configuring iPayment for a particular payment system using the iPayment administration user interface, you have to assign the payment system username and password in the Payment system configuration screens. You have to assign the same username and password in Apache Server.

For details on setting up basic authentication in Apache Server, refer to your Apache Server documentation.

IP Address Restriction

In addition to using the merchant username and password, you can restrict access to iPayment and payment systems through IP address restriction. By using IP address restriction, a feature of the Apache Server, you can specify one or both of the following parameters:

- The IP addresses of all trusted hosts (machines whose requests of iPayment the web server should accept)
- The IP addresses of some non-trusted hosts (machines whose request of iPayment the web server should refuse)

If a request is from a machine on the trusted list, iPayment processes the requested transaction. If the request is from a machine on the non-trusted list, Apache Server denies the request and prevents iPayment from processing it.

Through IP address restriction, you can limit access to all operations from non-trusted machines.

For more information about IP address restriction, including how to specify trusted hosts, see Apache Server documentation.

Data Privacy

iPayment provides data privacy. Sensitive data such as usernames, passwords, credit card, purchase card and bank account numbers are encrypted in the database.

Other Security Related Information

- iPayment is bundled with electronic commerce applications, executing as one application through the use of Java APIs. This prevents network transfer of sensitive information between electronic commerce applications and iPayment.
- Separate HTTP ports for site administration and iPayment administration increases security.
- Communication with back-end payment systems is via SSL.
- Bank account transfers are supported via account transfers interface. The payment system must be implemented using Java APIs directly and executed in the same process space to avoid network transfer of sensitive information between iPayment and payment systems.

Understanding Extensibility

For extensibility to work the customer has to implement the `oracle.apps.iby.extend.TxnCustomizer` interface. This interface has two methods: one method is called immediately before a request is sent to the BEP, and the other method is called immediately after the BEP sends a response. Each method is passed a three letter suffix identifying the BEP, a hashtable of name-value pairs comprising the transaction request/response, and an open database connection so that the custom parameters may be fetched/stored.

Extensibility will typically have the following workflow:

1. The electronic commerce application integrating with iPayment will first write custom BEP parameters to the database.
2. It will then send a transaction request to iPayment, during which the extensibility class implemented will query the custom parameters and add them to the request.
3. After a BEP response is generated, the extensibility class is again called and custom parameters sent by the BEP into the database are written. These parameters are queried later by the electronic commerce application or the extensibility class itself, which can use them for follow-on transactions.

2

Administering iPayment

This topic group provides process-oriented, task-based procedures for using the user interface to set up the application and perform essential business tasks.

Administration Overview

All setup and administrative functions of iPayment are done through the iPayment user interface. You can login and create, modify, and inactivate payment systems, payees, risk management properties, and routing rules.

iPayment is administered through a browser based administration user interface that is implemented using Java and Java Server Pages (JSP). Administering iPayment includes using the iPayment user interface to configure iPayment, to add and configure the payment systems, payees, routing rules, and risk management.

Note: Procedure for creating an iPayment administrative user is documented in *Oracle iPayment Implementation Guide.*

iPayment Administration User Interface

The following table lists the tab names and the functionality available from the iPayment administration user interface.

Table 2–1 Tabs and Functionality

Tab Name	Functionality
Payment System	Create and modify payment system properties in iPayment.

Table 2–1 Tabs and Functionality

Tab Name	Functionality
Payee	Create and modify payee properties and risk management properties in iPayment.
Routing Rule	Create, modify, and delete routing rules in iPayment.
Operation	View, create, and modify online operations in iPayment.
Properties	View and modify the values of iPayment properties.

Navigating the iPayment Administration User Interface

The iPayment administration user interface includes the administration tabs and the administration workspace.

The administration tabs on the top of the screen, remain visible as you navigate through iPayment. The tabs list the administrative tasks that you can perform. When you click a tab, details for the selected task appear in the administration workspace in the lower portion of the screen.

Payment System

You can perform the following tasks from the Payment Systems screen. This screen lists all the registered payment systems and links to each administration site.

- Creating a New Payment System
- Modifying a Payment System
- Updating a Default Payment System

Creating a New Payment System

Use this procedure to create and register a new payment system from the Create Payment System screen. The payment system is permanently registered.

Prerequisites

None

Steps

1. Click the Payment System tab. Payment Systems screen appears.
2. Click the **Create** button. Create Payment System screen appears.

3. Enter payment system details. Fields marked with red asteriks are mandatory fields. See Guidelines on Payment System Fields for a description of all fields.
4. Click **Create**.

Guidelines on Payment System Fields

The following table lists the payment system fields.

Table 2–2 Payment System Fields

Field/Checkbox Name	Description
* Name	Name of the payment system being added. This name is used in iPayment generated screens and reports, and should be a popularly known name of the payment system.
* Suffix	Three-character suffix for this payment system to use in iPayment API names (for example, *psa* for Payment System A). This has to be unique and is stored in lower-case.
User Name	Username that is to be used for authentication by the payment system when basic authentication is set up on the payment system servlet. The payment system username contains a maximum of 30 characters. This field is mandatory for credit card and purchase card payment instruments.
User Password	The password for the payment system username. This contains a maximum of 12 characters. This field is mandatory for credit card and purchase card payment instruments.
Base URL	Universal Resource Locator (URL) to invoke the payment system. The base URL should include a port number, if there is one. This generally starts with http://. However, if this URL is SSL enabled, then start with https://. This field is mandatory for credit card and purchase card payment instruments.
Administration URL	The URL that provides access to the payment system's administration tools. It is set up to start with either http or https. (This URL is also available in the Go To Vendor Site column on the Payment System Profiles screen.)
* Supported Payment Instruments	iPayment supports credit card transactions, purchase cards, and bank account transfers. Check the appropriate checkbox. For bank account transfers, enter the number of days, if there is a lead time to process a request.

Table 2–2 Payment System Fields

Field/Checkbox Name	Description
NLS Language	Preferred and optional languages and character sets supported by the payment system. These should be in Oracle NLS_LANG parameter format.

Modifying a Payment System

Use this procedure to modify the values associated with a payment system from the Payment System Details screen.

Prerequisites

None

Steps

1. Determine the payment system for which the properties are to be modified.
2. Click the payment system name from the Payment Systems screen. Payment System Details screen appears.
3. Modify the fields associated with a payment system. See Guidelines on Payment System Fields for field details.
4. Click **Update**.

Updating a Default Payment System

Use this procedure to update the default payment system per instrument type from the Default Payment System screen. Any transaction not routed by the routing rules is routed to the default payment system based on its instrument type.

Prerequisites

None

Steps

1. Click the Payment System tab.
2. Click the Default Payment System subtab. Default Payment System screen appears.

3. For each payment instrument, select a default payment system from the drop down list available in the Select Payment System column.
4. Click **Update**.

Payee

Payees and Risk Management screen displays a list of all the registered payees, their status, and their associated risk management links. You can perform the following tasks from the Payees and Risk Management screen.

- Creating a New Payee
- Modifying a Payee
- Inactivating a Payee
- Updating the Risk Management Status

Creating a New Payee

Use this procedure to add a new payee from the New Payee screen.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Create** button. New Payee screen appears.
3. Add payee details. Fields marked with red asteriks are mandatory fields. See Guidelines on Payee Fields for a description of all the fields.
4. Click **Create**. A new payee is created.

Guidelines on Payee Fields

The following table lists the payee fields.

Table 2–3 Payee Fields

Name	Description
* Payee Name	Payee name that appears on the pages and reports generated by iPayment. It is unique and case sensitive.
* Payee Identifier	iPayment uses this payee identifier to identify a particular payee. You cannot modify this identifier after saving it. It is unique, case sensitive and must contain alpha numeric and "_ ".
* Status	Payee status is either active or inactive. Select active if you want iPayment to process requests for this merchant. Select inactive to suspend payment processing for a merchant while maintaining access to the payee's configuration file.
Payment System Identifier	Identifier by which this payee is uniquely known to a payment system. It is provided by the payment system. Two payees cannot have the same identifier for one particular payment system. This is case sensitive.
Accepted Payment Instrument	Select the appropriate payment instrument that the payee accepts.

Modifying a Payee

Use this procedure to change a payee's properties from the Payee Details screen.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the payee name which appears in the Select Payee column. Payee Details screen appears.
3. Modify the fields associated with a payee. See Guidelines on Payee Fields for details.
4. Click **Update**.

Inactivating a Payee

Use this procedure to inactivate a payee from the Payee Details screen.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the payee name which appears in the Select Payee column. Payee Details screen appears.
3. Click Inactive in the Status radio button.
4. Click **Update**.

Updating the Risk Management Status

Use this procedure to update the risk management status of each payee from the Risk Management Status screen.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click either Enabled or Disabled link in the Risk Management Status column. Risk Management Status screen for that payee appears.
3. Click either **Enabled** or **Disabled** radio button.
4. Enter an integer value between 0 and 100 in the Cumulative Risk Threshold field.
5. Click **Update**.

References

For more information, "See Understanding Risk Management."

Risk Factors

You can modify the following risk factors and the risk score from the Risk Factors screen. All risk factors and the risk score use default values until the values are modified. After being modified, that particular factor or score only affects that particular payee. The other unchanged factors will continue to use the default values.

- Modifying the Risk Score
- Modifying the Payment Amount Limit Risk Factor
- Modifying the Payment History Risk Factor
- Modifying the Transaction Amount Risk Factor
- Modifying the Time of Purchase Risk Factor
- Modifying the AVS Codes Risk Factor
- Modifying the Frequency of Purchase Risk Factor
- Modifying the Oracle Receivables Risk Codes Risk Factor
- Modifying the Oracle Receivables Credit Rating Codes Risk Factor

For more information, See "Understanding Risk Management."

Modifying the Risk Score

Use this procedure to modify the risk score from the Risk Factors screen.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.
3. Select Risk Score from the Risk Factors list. Risk Score details appear.
4. Enter an integer value from 0 to 100 in the Risk Value column.
5. Click **Update**.

Modifying the Payment Amount Limit Risk Factor

Use this procedure to modify the payment amount limit risk factor from the Risk Factors screen. Payment amount is the amount involved in the payment request. Its scale varies from business to business. Based on the business model, each risk level varies with different amount ranges.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.
3. Select payment amount limit from the Risk Factor drop down list. Payment amount limit details appear.
4. For each risk level, enter a positive integer representing the lower bound of the amount range in Greater than or equal to column.
5. Click **Update**.

Modifying the Payment History Risk Factor

Use this procedure to modify the payment history risk factor by configuring the frequency values from the Risk Factors screen.

This risk factor tracks the reliability of the payor involved in a payment request. If a payor has a good history of payments over a long duration, then payments requested by this payor are considered to be low risk payments.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.

3. Select payment history from the list available in the Risk Factor list. Payment history details appear.
4. Enter a positive integer for the duration value and select a duration time.
5. For each risk level, enter a positive integer representing the lower bound of the frequency range in Greater than or equal to column.
6. Click **Update**.

Modifying the Transaction Amount Risk Factor

Use this procedure to modify the transaction amount risk factor from the Risk Factors screen. Transaction amount is the total amount of payments made using the same instrument in a specified period of time. If the total payment amount exceeds the transaction amount, the payment is considered highly risky.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.
3. Select transaction amount from the Risk Factor drop-down list. Transaction amount details appear.
4. Enter a positive number in the amount field, a positive integer in the duration field and select a duration time unit.
5. Click **Update**.

Modifying the Time of Purchase Risk Factor

Use this procedure to modify time of purchase risk factor from the Risk Factors screen.

Time of purchase is the time at which a payment request is made by the payee's customer. A risk level can be associated to every hour of the day. No hour can be associated with more than one risk level.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.
3. Select Time of Purchase from the Risk Factors drop-down list. Time of Purchase details appear.
4. For each time range, select the starting and ending hour and its risk level.
5. Click **Update**.

Guidelines

You can add more time ranges and the risk levels associated with them by entering time ranges and risk levels in the last row of the table. You can also delete time ranges by selecting the corresponding checkbox in the Remove column. No time ranges should overlap.

Modifying the AVS Codes Risk Factor

Use this procedure to modify the AVS codes risk factor from the Risk Factors screen.

AVS codes is returned by the payment systems such as Cybercash and others. Address verification service codes are provided by Mastercard and Visa credit card networks to match the billing address with the address that is maintained for the cardholder by the issuing bank. These codes can be associated with various risk levels.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.
3. Select AVS codes from the Risk Factor list. AVS codes details appear.
4. Enter the AVS codes (separated by commas), in the address verification service codes field, corresponding to each risk level.
5. Click **Update**.

Guidelines

If you remove all existing AVS codes, iPayment restores the default values.

Modifying the Frequency of Purchase Risk Factor

Use this procedure to modify the frequency of purchase risk factor from the Risk Factors screen. This factor is used to track sudden surge in the use of a payment instrument in a payment request. If the frequency of use of an instrument in a duration configured is more than the setup value, then the payment request is considered to be a high risk payment.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.
3. Select Frequency of Purchase from Risk factors list. Frequency of Purchase details appear.
4. Enter a positive integer for the maximum frequency of payment and in the duration field. Also enter the duration time period.
5. Click **Update**.

Modifying the Oracle Receivables Risk Codes Risk Factor

Use this procedure to modify Oracle receivables risk codes risk factor from the Risk Factors screen. Risk code is a user defined risk assessment field in Oracle Receivables. It is useful for online financing or for evaluating purchases of a large amount for a new customer. A payee associates risk levels with each risk code.

Prerequisites

- Install and register Oracle Receivables.

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.

3. Select Oracle Receivables Risk Codes from the Risk Factors list. Oracle Receivables Risk Codes details appear.
4. Select risk levels corresponding to Oracle Receivables Risk Codes in each row.
5. Click **Update**.

Modifying the Oracle Receivables Credit Rating Codes Risk Factor

Use this procedure to modify Oracle Receivables Credit Rating Codes risk factor from the Risk Factors screen. Credit Rating is the information enabling payees to effectively manage financial terms with their customers. It is useful for online financing or for evaluating purchases of large values for a new customer. A payee associates risk levels to each credit rating.

Prerequisites

1. Install and register Oracle Receivables.

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click the **Modify** button in the Risk Factors column associated with the payee whose risk factor is to be configured. Risk Factors screen appears.
3. Select Oracle Receivables Credit Rating Codes from the Risk Factors list. Details appear.
4. Select Risk Levels corresponding to Oracle Receivables Credit Rating Codes in each row.
5. Click **Update**.

Modifying the Risky Instruments Risk Factor

This risk factor cannot be configured.

Risky instruments are a list of instruments that are considered risky by each payee. These include the instruments that were used by customers earlier and had resulted in fraud or chargebacks. If the instrument being used for payment is found in the repository, the payment is considered a high risk payment.

Modifying the Ship to/Bill to Address Risk Factor

This risk factor cannot be configured.

Ship to / bill to address is used to match ship to and bill to addresses in the payment request. If the ship to and bill to addresses do not match, the payment request is considered high risk.

Modifying the Oracle Receivables Transactional Credit Limit Risk Factor

This risk factor cannot be configured.

Transaction credit limit is the credit limit per transaction assigned by Oracle Receivables. When a payment request exceeds the transaction credit limit, it becomes a risky payment. It varies from business to business and can be set up at customer or site level through Oracle Receivables.

Modifying the Oracle Receivables Overall Credit Limit Risk Factor

This risk factor cannot be configured.

Credit limit is an overall credit limit assigned at site level. If a customer has an outstanding balance and the total amount of payment made by the customer exceeds the overall credit limit, the payment becomes a high risk payment. It varies from business to business and can be set up at customer or site level through Oracle Receivables.

Risk Formula

You can perform the following procedures from the Risk Formula screen. This screen lists all the risk formulas available for the selected payee. Every payee created through the administrative user interface generates an implicit risk formula associated with that payee. The implicit risk formula cannot be deleted. It is

generated with equal weights among the default risk factors. The weights for an implicit risk formula can be changed like weights for any other formula.

- Creating a Risk Formula
- Updating a Risk Formula
- Deleting a Risk Formula

References

For more information, See Understanding Risk Management.

Creating a Risk Formula

Use this procedure to create a risk formula from the Risk Formula screen.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click **Create/Modify** available below the Risk Formulas column associated with the payee.
3. Risk Formulas screen appears. The screen lists the implicit formula for each payee and other risk formulas of this payee. Click **Create**. New Risk Formula screen appears.
4. Enter a unique name for the new risk formula in the Formula Name field.
5. Enter a positive integer weight in percent for each risk factor. The total weight of all risk factors should be equal to 100. If Oracle Receivables is installed on your site, Oracle Receivables risk factors also appear on this screen.
6. Click **Create**.

References

For more information, See Understanding Risk Management.

Updating a Risk Formula

Use this procedure to update risk formula from the Risk Formula screen.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click **Create/Modify** available below the Risk Formulas column associated with the payee.
3. Select the risk formula to be modified. Click the name of the formula. Risk Formula Details screen appears listing the risk factors and the weights assigned to each of the risk factors.
4. Enter a positive integer weight in percent for each risk factor. The total weight of all risk factors should be equal to 100. If Oracle Receivables is installed on your site, Oracle Receivables risk factors also appear on this screen.
5. Click **Update**.

References

For more information, See Understanding Risk Management.

Deleting a Risk Formula

Use this procedure to delete risk formula, except the implicit risk formula, from the Risk Formula screen.

Prerequisites

None

Steps

1. Click the Payee tab. Payees and Risk Management screen appears.
2. Click **Create/Modify** available below the Risk Formulas column associated with the payee. Risk Formulas screen appears. The screen lists the implicit risk formula for each payee and other risk formulas for this payee.

3. Check the check box available in the Remove column corresponding to the risk formula which is to be deleted.
4. Click **Update**.

Routing Rule

You can perform the following tasks from the Routing Rules screen.

- Changing the Routing Rule Priority
- Creating Routing Rules
- Modifying Routing Rules
- Deleting Routing Rules

References

For more information, See iPayment Routing and Operation.

Changing the Routing Rule Priority

Use this procedure to change the routing rule priority from the Routing Rules screen.

Prerequisites

None

Steps

1. Click the Routing Rule tab. Routing Rules screen appears.
2. Select the new priority for changing the routing rule from the drop down list in the Rule Priority column. All routing rules are automatically reordered in the database.

Creating Routing Rules

Use this procedure to create Routing Rules from the Create Routing Rule screen.

Prerequisites

None

Steps

1. Click Routing Rule tab. Routing Rules screen appears.
2. Click the **Create** button. Create Routing Rule screen appears.
3. Enter values in the fields on this screen. Fields marked with red asteriks are mandatory. See Guidelines on Routing Rule Fields for a description of all the fields.
4. Click **Create**.

Guidelines on Routing Rule Fields

The following table lists the routing rule fields.

Table 2–4 Routing Rule Fields

Name	Description
* Rule Name	Name of the Routing rule. This name must be one word without spaces and must be unique. The name should contain alpha numeric characters and _ only.
* Rule Priority	Select the new priority from the drop down list.
* Status	Select the status of the Routing Rule as either Active or Inactive.
* Route to Payment System	Select the payment system to which transactions that satisfy the conditions are routed.
Rule Condition	Atleast one rule condition should be enabled.
* Amount	This is a rule condition. Select the desired operation and enter an amount value.
* Instrument type	This is a rule condition. Select an operation and an instrument type.
* Currency Type	This is a rule condition. Select an operation and a currency type.

Modifying Routing Rules

Use this procedure to modify routing rules from the Routing Rule screen.

Prerequisites

None

Steps

1. Click the Routing Rule tab. Routing Rule screen appears.
2. Click the routing rule name which appears below the Select Rule column. Routing Rule Details screen appears with all the routing rule fields. See Guidelines on Routing Rule Fields.
3. Modify the fields.
4. Click **Update**.

Guidelines

For more information, See iPayment Routing and Operation.

Deleting Routing Rules

Use this procedure to delete routing rules from the Routing Rule screen.

Prerequisites

None

Steps

1. Click the Routing Rule tab. Routing Rules screen appears.
2. Select the routing rule to be deleted by checking the corresponding checkbox in the Remove column.
3. Click **Update**.

Managing Operations

A user interface is provided in iPayment to test authorization and capture operations for online processing of credit cards and purchase cards. The user interface can also be used to search and view details on actual transactions which have been submitted through iPayment.

Through the user interface, a test authorization request can be submitted by supplying transaction details such as the credit card number, payee, and amount. Risk management details can be supplied to enable risk analysis on the transaction. Once the transaction is submitted, the results of the authorization operation are returned.

Note: The Operations user interface should be used by implementers of the Oracle iPayment software to test iPayment set-up. It should not be used for processing real payments.

The user interface can also be used to view and search on all transactions which have been processed by iPayment. Transactions can be searched by various search criteria. Transactions matching the criteria are displayed in a summary format. More details on a transaction can viewed by selecting the transaction from this summary list. iPayment user interface also provides the ability to submit follow-on operations from the search screens.

Performing Payment Authorization

Use this procedure to perform a test on payment authorization from the Authorization Details screen. This procedure can be used to test if the payment system, payee, risk factors, risk formulas, and routing rules are all setup correctly within iPayment

Prerequisites

1. Setup the Payment System, Payee, and Routing Rules during iPayment installation.
2. Setup the Risk Factors and Risk Formulas, if you are also testing Risk Analysis.

Steps

1. Click the **Operation** tab. Authorization Details screen appears.

2. Enter the Authorization details and click **Next.** Authorization Summary screen appears with the details entered in the Authorization Details screen. If you are satisfied with the information entered, click **Submit**. See Guidelines on Authorization Details Fields.

3. Authorization Results screen appears with the entire response from the system about the Authorization success or Authorization failure. Click **Done**, to complete the Authorization operation.
   If you want to make changes to the selections made in case of a failed operation, click the **Back** button to navigate to the first Authorization Details screen.

Guidelines on Authorization Details Fields

The following table lists the authorization details fields.

Table 2–5 Authorization Details Fields

Name	Description
Payee	The list contains the payees configured in iPayment. The selected payee would be the one receiving payment from this operation.
Tangible ID / Order ID	The identifier for this operation that corresponds to a particular order. This field with the payee field must be unique.
Amount	The authorization amount in the format matching the currency code. The amount must be a positive value.
Currency	The currency used for this operation.
Payment Instrument Type	Either credit card or purchase card. These are the only two payment instruments supported through the iPayment Operations user interface.
Retry	Select Yes or No button. Select Yes, if iPayment did not return a valid response for a previous attempt of the same transaction.
Authorization Type	Select an authorization type from the list. Authorization Only operation performs only the authorization operation. Authcapture operation performs both authorization and capture operations. Check the payee's setup to decide which authorization type is configured for the payee.
Voice Authorization	Select either Yes or No button. Select Yes, if an authorization has already been made by contacting the financial institution directly rather than through iPayment, and only if you have an Authorization Code for that authorization issued by the financial institution.

Table 2–5 Authorization Details Fields

Name	Description
Authorization Code	If an authorization has already been made by contacting the financial institution directly, rather than through iPayment, and the financial institution has issued an authorization code for that transaction, enter the authorization code in this field and select Yes for Voice Authorization (see above).
Purchase Order Number	This field appears on the user interface if the payment instrument is a purchase card. It is a number from the payor for this authorization operation.
Tax Amount	This field appears on the user interface if the payment instrument is a purchase card. The amount of the authorization that is taxable.
Ship to Zip Code	This field appears on the user interface if the payment instrument is a purchase card. The zip code of the destination where the physical merchandise would be shipped.
Ship From Zip Code	This field appears on the user interface if the payment instrument is a purchase card. The zip code of the source from where the physical merchandise is shipped.
Card Number	The payment instrument card number used for authorization.
Expiration Date	The expiration date corresponding to the card number of the instrument. The date the card expires or can no longer be used.
Card Holder Name	The name that appears on the card corresponding to the payment instrument.
Card Type	The card type of the payment instrument. Choose Unknown for the card types not in the list.
Card Subtype	This field appears on the user interface if the payment instrument is a purchase card. The subtype of the purchase card. Choose Unknown for the card types not in the list.
Street	The number and street name of the customer's billing address.
City	The name of the customer's city in the billing address.
State	The short code for the customer's state in the billing address.
Zip Code	For customers in the United States, the zip code.
Country	The name of the customer's country in the billing address.
Risk Analysis	Select yes or No radio button. Select yes if risk analysis is to be performed on this transaction.

Table 2–5 Authorization Details Fields

Name	Description
Risk Formula Name	Select a risk formula from the list. It is a formula used to calculate the risk score. This field is only saved if you select yes for risk analysis.
Same Shipping and Billing Address	Select Yes if the customer's shipping and billing addresses are the same. This field is only saved if you select yes for risk analysis.
Time of Purchase	Select a time of the day at which this operation is performed. By default the time of the current system appears in this field. This field is only saved if you select yes for risk analysis.
Customer Account Number	The Oracle Accounts Receivables account number for the customer. This account number is used to retrieve the Oracle Accounts Receivables risk factors.This field is only saved if you select yes for risk analysis.

Searching Transactions

You can search for transactions based on one or more criteria in the Search Transactions screen. To narrow the search criteria, you must enter as many search criteria as possible. Enter values in the fields on which you want to perform the search.

You can also perform Capture and Re-Authorization follow-on operations on the Matching Transactions screen.

Steps

1. Click the Operation tab. Authorization Details screen appears.
2. Click the Follow-On Operation subtab. Search Transactions screen appears. See Guidelines on Search Transaction Fields
3. Select a payee from the list of registered payees.
4. Enter any other additional search criteria.
5. Click **Go**. Matching Transactions screen appears with the details depending on the search criteria entered.

Guidelines on Search Transaction Fields

The following table lists the Search Transaction Fields.

Table 2–6 Search Transaction Fields

Field	Description
Payee	The list contains payees configured in iPayment that have received payments during previous operations.
Tangible ID / Order ID	Identifiers used for previous orders. This field is case sensitive. The % sign can be used as a wild card.
Starting Date / Ending Date	Date ranges for dates of operations.
Transaction Status	Current status of each transaction. You can select more than one value for this field. See transaction status diagram for details.

Matching Transactions

After performing a search, Matching Transactions screen displays summary information for transactions matching the search criteria.

Prerequisites

- Complete the initial steps for searching a transaction on the Search Transactions screen.

Steps

1. Click the hypertext value in the TangibleID / OrderID field for which you want to view the details. Transaction Summary screen appears with the details. Click **Done**.

> **Note:** Transactions can be sorted either by Date(Ascending / Descending) or by Tangible ID / OrderID. Default sorting is done by TangibleID field.

2. Alternatively, you can perform follow-on operations by selecting the link in the Follow-on operations column.

Performing a Capture Operation

You can perform a capture operation starting at the Matching Transactions screen.

Prerequisites

1. Identify the transaction for which the Capture operation is to be performed.
2. The transaction supports Capture as a Follow-On operation.

Steps

1. Click **Capture** in the Follow-On operations column on the Matching Transactions screen. Capture Details screen appears. By default the Amount field contains the Authorized amount.
2. If you want to perform the capture operation for an amount different than the authorized amount, edit the **Amount** field and click **Submit**.
   Capture Results screen appears with the status of the capture operation, the Transaction Date, and the Transaction Type.

Note: The Amount value should be less than or equal to the Authorized amount.

Viewing Transaction Summary

You can view the entire history of a transaction from the Transaction Summary screen.

Steps

1. Click TangibleID / OrderID on the Matching Transactions screen to review the Transaction Details.
2. Transaction Summary screen appears displaying the transaction details for a transaction. Click **Done** to return to the Matching Transactions screen.
3. For more details on a particular transaction, click the Trxn Type link. Trxn Details screen appears with the details.

Guidelines on Transaction Details Fields

The following table shows the fields in the Transaction Details screen.

Table 2–7 Table on Response Fields

Name	Description
Payee	The identifier of the payee who received payment in the operation.
Tangible ID / Order ID	The identifier for this operation that corresponds to a particular order.
Amount	The amount for this authorization amount in the format matching the currency code.
Currency	The currency used for this operation.
Authorization Code	The authorization Code.
AVS Code	Address Verification Service Code.
Transaction Date	The date when the transaction is processed.
Transaction Type	The type of the transaction.
Credit card Number / Purchase card Number	The credit card or purchase card number used for making payments. For security, only the first four digits of the card number are shown during the transaction review.
Card Type	The type of credit card or purchase card.
Auxiliary Message	Additional message from the processor.
Reference Code	The retrieval reference number.
Overall Risk Score	The overall score from risk evaluation.
Risk Threshold	The value set up by the payee to check against overall risk score.
Risky Transaction	The value in this field is true or false depending if the transaction is risky or not.
Payment System Name	The name of the payment system that processed the transaction.
Error Location	Error location reported by the payment system. It is only present when an error has occurred on the payment system's side.
Payment System Code	Error Code reported by the payment system. It is only present when an error has occurred on the payment system's side.

Table 2–7 Table on Response Fields

Name	Description
Payment System Message	Error message reported by the payment system. It is only present when an error has occurred on the payment system's side.

iPayment Properties

You can use the Property Manager provided by the Admin Console to set up or modify iPayment properties.

All the properties listed below are mandatory and should not be removed or renamed, however their values can be modified.

Note: There are two more properties on the Admin Console. The properties are service.factories and service.oracle.apps.iby.ecapp.PaymentServiceFactory.desc. Do not modify the values of these properties as these are code specific and are needed by iPayment.

iPaymentURL

This property contains the following URL:

http://machine:port/<jsp>/ecapp?

Replace the machine and port with the names of the actual machine and the actual port where the iPayment ECServlet is installed. Also, make sure that "?" is present at the end of the URL or append "?" at the end.

This information is mandatory if your electronic commerce applications use iPayment PL/SQL APIs or if your application is an Oracle 3i client.

Errorfile

This property specifies the fully qualified name of the error file generated by iPayment. For example: <path>/iby-error.log. Make sure you give the absolute path to your logging directory. The error file contains exceptions and error messages. It is always generated regardless of the debug flag.

Debugfile

This property specifies the fully qualified name of the debug file generated by iPayment. For example,<path>iby-debug.log. Make sure you give the absolute path to your logging directory. This debug contains debug messages and error messages. It is generated only when the debug flag is turned on.

Debug

This property is either true or false. If it is set up to true, iPayment writes debug messages to the Debugfile. The default value is false.

Http_Proxy

This property specifies the proxy-URL. For example, http://www-proxy.us.oracle.com.

To set up this property with an empty value, insert a string starting with <. For example, <none>.

No_Proxy

This property specifies the domain name for which no proxy is needed. For example, us.oracle.com.

To set up this property with an empty value, insert a string starting with <. For example, <none>.

Modifying iPayment Properties

Modify the following iPayment properties to ensure that PL/SQL API and error functionality work.

iPaymentURL

This property contains the following URL:

http://machine:port/jsp/ecapp?

Replace the machine and port with the names of the actual machine and the actual port where the iPayment ECServlet is installed. Also, make sure that "?" is present at the end of the URL or append "?" at the end.

This information is mandatory if your electronic commerce applications use iPayment PL/SQL API.

Errorfile

This property specifies the fully qualified name of the error file generated by iPayment. For example: <your_log_dir>/iby_error.log. Make sure you give the absolute path to your logging directory. The error file contains exceptions and error messages. It is always generated regardless of the debug flag.