Oracle® iPayment

Concepts and Procedures

Release 11*i*

Part No. A95477-06

May 2005

ORACLE®

Oracle iPayment Concepts and Procedures, Release 11*i*

Part No. A95477-06

Contributors: Jill Burton, Jonathan Leybovich, Rajiv Menon, Elizabeth Newell, Aalok Shah, Ramesh Shankar, Feiyun Zhang, Ramasubramanian Balasundaram

Contents

2 Administering iPayment

3 Understanding Integration with Oracle Payables

4 Transaction Reporting

Index

Send Us Your Comments

Oracle iPayment Concepts and Procedures, Release 11*i*

Part No. A95477-06

Oracle 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.

- Electronic mail: appsdoc_us@oracle.com
- FAX: (650) 506-7200 Attention: Oracle Applications Documentation
- Postal service:
  Oracle Corporation
  Oracle Applications Documentation
  500 Oracle Parkway
  Redwood Shores, CA 94065
  USA

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 Release 11*i* of the Oracle iPayment Concepts and Procedures Guide.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Oracle iPayment

  If you have never used Oracle iPayment, Oracle suggests you attend one or more of the Oracle iPayment training classes available through Oracle University.
- The Oracle Applications graphical user interface.

  To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User’s Guide*.

See Other Information Sources for more information about Oracle Applications product information.

How To Use This Guide

The Oracle iPayment Concepts and Procedures Guide contains the information you need to understand and use Oracle iPayment. This guide contains four chapters:

- Chapter 1, "Understanding iPayment"

  This chapter 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.

- Chapter 2, "Administering iPayment"

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

- Chapter 3, "Understanding Integration with Oracle Payables"

  This chapter provides details on the integration of iPayment and Oracle Payables.

- Chapter 4, "Transaction Reporting"

  This chapter provides details of the pages provided for viewing the key performance metrics such as transaction summaries, payee summaries, and other critical performance indicators.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/ .

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Other Information Sources

You can choose from many sources of information, including documentation, training, and support services, to increase your knowledge and understanding of Oracle iPayment.

If this guide refers you to other Oracle Applications documentation, use only the Release 11*i* versions of those guides.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF).

- **PDF Documentation-** See the Online Documentation CD for current PDF documentation for your product with each release. This Documentation CD is also available on Oracle*MetaLink* and is updated frequently.
- **Online Help -** You can refer to Oracle Applications Help for current HTML online help for your product. Oracle provides patchable online help, which you can apply to your system for updated implementation and end user documentation. No system downtime is required to apply online help.
- **Release Content Document -** See the Release Content Document for descriptions of new features available by release. The Release Content Document is available on Oracle*MetaLink*.
- **About document -** Refer to the About document for information about your release, including feature updates, installation information, and new documentation or documentation patches that you can download. The About document is available on Oracle*MetaLink*.

Related Guides

Oracle iPayment shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other guides when you set up and use Oracle iPayment.

You can read the guides online by choosing Library from the expandable menu on your HTML help window, by reading from the Oracle Applications Document Library CD included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle Store at http://oraclestore.oracle.com.

Guides Related to All Products

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Guides Related to This Product

Oracle Payables User Guide

This manual describes how accounts payable transactions are created and entered into Oracle Payables. This manual also contains detailed setup information for Oracle Payables and discusses suppliers, banks, invoices, and also explains how to create payments and run reports.

Oracle Receivables User Guide

This manual describes how accounts receivables transactions are created and entered into Oracle Receivables. This manual also contains detailed setup information for Oracle Payables and discusses customers, banks, invoices, and reporting.

Oracle iReceivables Implementation Guide

This manual describes the setup tasks that you need to perform for iReceivables and information you need to configure iReceivables to suit your business requirements.

Oracle Cash Management User Guide

This manual explains how you can reconcile your payments with your bank statements.

Oracle Collections User Guide

This manual explains the key features and process flows in Collections.

Oracle iStore Implementation and Administration Guide

This manual explains the information needed to implement, administer, and maintain Oracle iStore.

Oracle Order Management User Guide

This manual explains how enter, maintain, and process orders and returns.

Installation and System Administration

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11*i*. It provides a useful first book to read before an installation of Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Business Intelligence (BIS), languages and character sets, and Self-Service Web Applications.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications and the Oracle technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user guides and implementation guides.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11*i*. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11*i*. You cannot upgrade to Release 11*i* directly from releases prior to 10.7.

Maintaining Oracle Applications

Use this guide to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the Oracle applications file system and database.

Oracle Applications System Administrator’s Guide

This guide provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage concurrent processing.

Oracle Alert User’s Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Applications Developer’s Guide

This guide contains the coding standards followed by the Oracle Applications development staff and describes the Oracle Application Object Library components that are needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. This manual also provides information to help you build your custom Oracle Forms Developer forms so that the forms integrate with Oracle Applications.

Oracle Applications User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built by using Oracle Forms.

Other Implementation Documentation

Oracle Applications Product Update Notes

Use this guide as a reference for upgrading an installation of Oracle Applications. It provides a history of the changes to individual Oracle Applications products between Release 11.0 and Release 11*i*. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle Applications-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide

This guide describes how Oracle Applications users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup and reference information for the Oracle iPayment implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This guide also provides information on creating custom reports on flexfields data.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Oracle*MetaLink*

Oracle Self–Service Web Applications Implementation Manual

This manual contains detailed information about the overview and architecture and setup of Oracle Self–Service Web Applications. It also contains an overview of and procedures for using the Web Applications Dictionary.

Oracle Order Management APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes APIs and open interfaces found in Oracle Order Management Suite.

Other Information Sources

For more information, see the latest versions of the following manuals:

- *Oracle iPayment Implementation Guide*
- *iPayment JavaDoc* (Available on Oracle*MetaLink*)
- Apache Server Documentation (http://www.apache.com)
- Apache’s mod-ssl documentation (http://www.mod-ssl.org/docs)
- Java Developer’s Guide (http://www.sun.com)

Training and Support

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle iPayment and reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many education centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle iPayment working for you. This team includes your technical representative, account manager, and Oracle’s large staff of consultants and support specialists with expertise in your business area, managing an Oracle server, and your hardware and software environment.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

About Oracle

Oracle develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources and customer relationship management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and applications products, along with related consulting, education, and support services, in over 145 countries around the world.

Your Feedback

Thank you for using Oracle iPayment and this user guide.

Oracle values your comments and feedback. In this guide is a reader’s comment form that you can use to explain what you like or dislike about Oracle iPayment or this user guide. Mail your comments to the following address or call us directly at (650) 506-7000.

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Or, send electronic mail to appsdoc_us@oracle.com.

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.

Overview of Oracle iPayment

Oracle iPayment is a framework that lets you build integrations with financial institutions and payment processors for payment and receipt processing. In addition, a number of certified integrations are provided out of the box.

For inbound payments, Oracle iPayment supports four electronic payment methods: credit card, purchase card, PINless debit cards, and EFT transactions. For outbound payments, Oracle iPayment supports checks, wires, and EFT transactions. Oracle iPayment's support of checks refers to sending a payment instruction to your financial institution, not printing physical checks. Oracle iPayment also supports payment partner integrations, such as with Paymentech.

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

Key Benefits of iPayment

- Supports both outbound (payables) and inbound (receivables) payments. The ability to further consolidate the connections between your applications and your financial institutions helps reduce the cost of ownership.
- Operates simultaneously with multiple payment processing systems which lets businesses offer several payment options to their customers and reduces implementation and maintenance costs.
- Provides security through support for industry standards such as Secure Socket Layer (SSL).
- Integrates with other Oracle Applications, such as Oracle iStore through Order Capture and Order Management, Oracle Receivables, and provides a single, 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 (EC) or client-server applications. Oracle iPayment also lets both stand-alone businesses and internet service providers offer electronic payment processing.

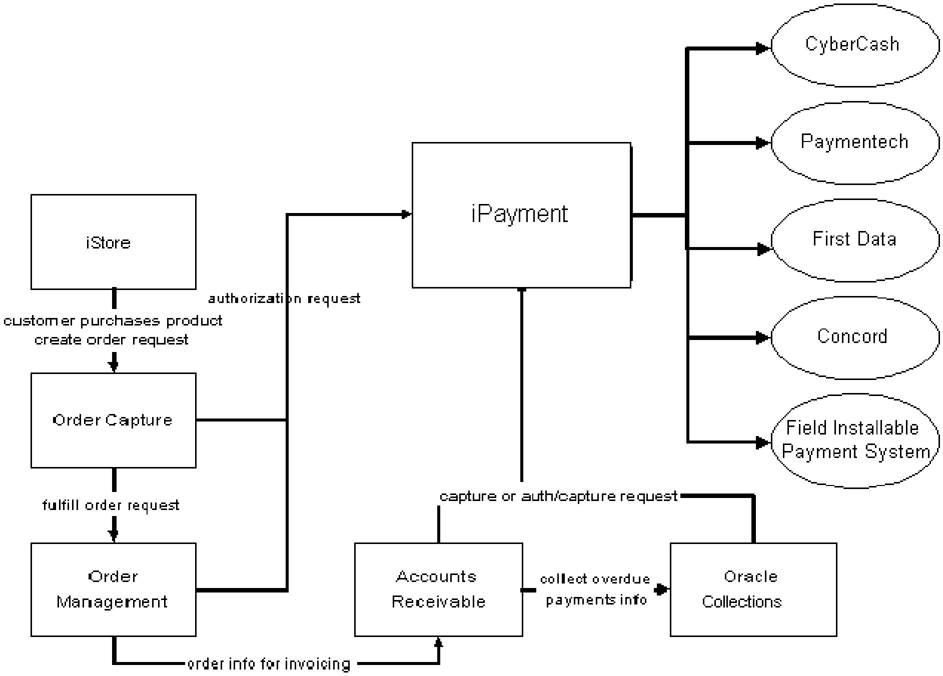
Integration with Other Oracle Applications

Oracle iPayment integrates with other Oracle Applications to provide payment processing across your enterprise. Various applications send payment transaction requests to Oracle iPayment for processing. Without Oracle iPayment, each of these applications would need to build integrations to back-end payment (BEP) systems. Oracle iPayment saves integration effort by providing a single source to the back-end payment systems such as Citibank, Concord EFSnet, First Data North, and other 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)**: A customer purchases a 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. Both use iPayment to verify if the credit card number is valid and authorize the order amount. They can also perform some risk evaluation as part of the authorization.
3. **Oracle Receivables:** When the order is shipped, the credit card information is passed to Oracle 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.

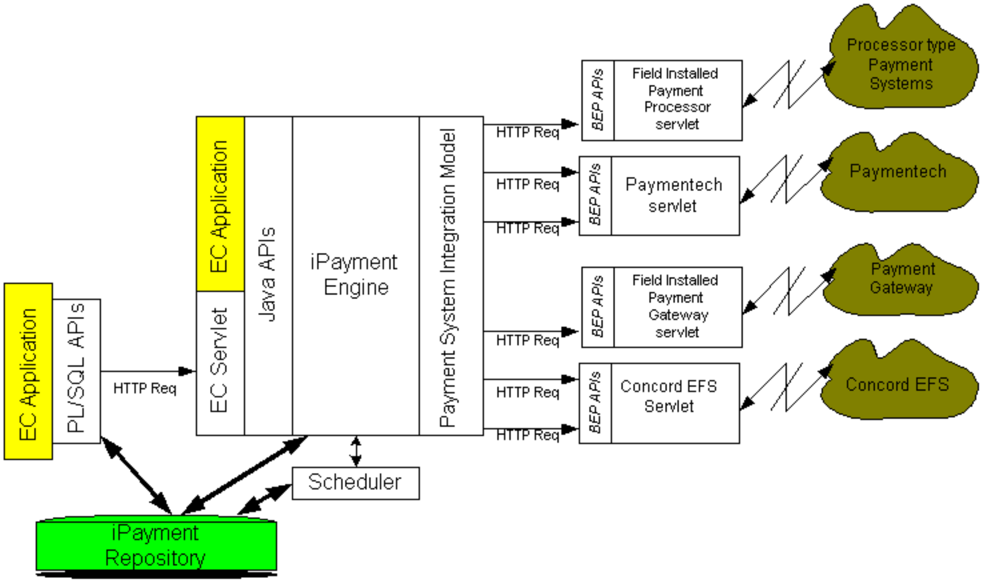
Figure 1–1 iPayment's Integration with Oracle Applications



iPayment Architectural Overview

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

Figure 1–2 iPayment Architecture



iPayment APIs

iPayment provides two types of APIs to simplify the integration of existing or new applications with iPayment for payment processing.

- **Electronic Commerce APIs**
  EC applications can use these APIs to integrate their applications with iPayment. The EC 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 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 provides the Payment System Integration Model to interface with payment gateways and payment processors.

iPayment Engine

The iPayment engine contains functionality for multi-payment method support, routing, risk management, and so on. 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, and return. This servlet is primarily used for the PL/SQL APIs provided by iPayment.

- Payment system servlets

iPayment bundles payment system servlets developed by Oracle and/or interfaces with servlets developed by its *payment system partners*. The payment systems communicate with the payment processors and the acquirers or banks to process payment transactions. iPayment includes payment system servlets for Paymentech, CyberCash, Citibank, First Data (North), and Concord EFSnet. Some payment systems, such as VeriSign, have built their own iPayment servlets.

- Field-installable servlets

iPayment supports *field-installable servlets*. These payment system servlets are 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 EC applications to customize the payment system for their specific needs and regions.

Field-installable payment system servlets for iPayment are available from Oracle’s payment system partners, or can be custom built.

Understanding Payees

In Oracle iPayment, the payee represents a first party entity that is collecting money from customers (in the case of inbound payments) or disbursing money to suppliers (in the case of outbound payments). A payee also has a business relationship with a payment system, in which the payment system processes transactions for the payee.

You can have more than one payee in Oracle iPayment for inbound and outbound payments. You can have multiple payees, for example, because different business units or legal entities in one organization want to process transactions through different payment systems, or use separate relationships with a payment system.

When you create a payee in Oracle iPayment, you must specify several pieces of information.

- An identifier that calling applications can use to identify the payee that the transaction belongs to.
- A list of payment instrument types that the payee supports
- Payment system identifiers that link the payee to payment systems.

Advanced features such as risk management, routing rules, and security are set up per payee, allow each payee to control the way transactions are processed.

Understanding Credit Card Transactions

Among inbound payments, Oracle iPayment handles credit card, PINless debit card, purchase card, and EFT transactions. This section 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 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 available credit to cover the transaction. If so, then the processor reserves the funds and sends back an authorization code.

- Settlement

  The merchant delivers goods to the customer and needs to capture the funds reserved in the authorization, which can occur at the same time as authorization. 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 for gateway-model and processor-model payment systems.

Understanding Gateway-Model and Processor-Model Payment Systems

Oracle iPayment supports both gateway-model and processor-model payment systems. The processor model describes the interface between Oracle iPayment and a payment processor. A payment processor is a service that interacts directly with banks and card institutions such as Visa, Mastercard, and American Express, to process financial transactions. The gateway model describes the interface between Oracle iPayment and a payment gateway. A payment gateway is a service provider that acts as an intermediary between a merchant and a payment processor.

A gateway-based system takes all transactions online. A processor-based system allows authorizations in real-time and follow-up transactions such as captures and credits offline. Offline transactions must be batched together and sent as a single request to the payment system. All transactions other than authorizations are, by default, performed offline. Offline transactions are sent to the processor when the next batch close operation is attempted.

You can do a batch close operation either manually or automatically. In a manual batch close, a call is made to the Oracle iPayment close batch API. The Oracle iPayment scheduler performs an automatic batch close. To determine the final statuses of all submitted transactions in a batch close, a follow-up call to the batch query API can be made. The follow-up call may be a manual call to the API, or can be made automatically through the Oracle iPayment scheduler. The follow-up call must be made after the batch is submitted. The actual period of time depends on the processor and the number of transactions in the batch.

The choice of integrating to a gateway-model or processor-model payment system is generally determined by the business type, number of transactions per day, and the acquiring bank. Processors typically have more rigorous security, connectivity, and testing requirements. Gateways provide ease-of-use, often using SSL based internet connectivity. Gateways charge additional fees (including per-transaction fees), beyond what the processor charges. Typically, pricing varies by payment system and the processor model payment systems often favors higher-volume merchants who are willing to put forth the effort and cost of processor connectivity. The Gateway model favors lower-volume merchants, or merchants who are willing to pay a per-transaction premium for easier set up and connectivity.

Understanding Terminal-Based and Host-Based Merchants

For gateway-model payment systems, Oracle iPayment supports these processing models that the financial industry uses for credit card transactions:

- Terminal-Based Merchant

  The payee or business determines when to close batches of transactions for clearing and settlement, and is responsible to perform the close batch operations.

- Host-Based Merchant

  In this model, the payment system's host machine maintains all the transactions and is usually responsible for close batch operations 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.

> **Note:** Processor model payment systems do not support host-based merchants.

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 that you choose affects how you perform the settlement operations. For a terminal-based merchant model, you must periodically perform close batch operations. Consult your acquiring bank for more information when you sign up.

Understanding Purchase Cards

Purchase cards, also known as procurement cards, are a special type of credit card that possess more features, capabilities, and controls than standard consumer credit or charge cards. Purchase cards are issued by an organization (hereafter referred to as buyer) to its employees. The card is generally used by the employees for purchasing corporate supplies and services. Payments are directly made by the corporate buyer to the card issuer. The difference is that for regular credit cards, payments are made by the individual buyer (who may be an agent of a corporate buyer) to the card issuer.

Central billing to the buyer, to which the cards are issued, is done. The merchant receives payment a few days after submitting a transaction and the buyer pays the issuing bank for the aggregate amount of purchases made in the billing period.

A purchase card transaction can contain level II or level III data. For more information on card data levels, see Purchase Card Data Levels. In a typical business-to-business scenario, level III data is used. Purchase cards provide merchants with a mechanism to eliminate the costly paper process of providing and collecting funds for outstanding invoices.

Oracle iPayment supports purchase cards requiring level II and level III data. Level III data is supported only for payment processors. From Oracle iPayment's perspective, purchase cards are similar to credit cards, except that the payment processor gets more information for purchase cards.

iPayment only supports Level III data purchase card transactions sent via Oracle Receivables. For more information, see the credit card chapter in the *Oracle Receivables User Guide*.

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 prefer merchants that accept purchase cards.
- Merchants generally receive better rates with purchase cards than with credit cards.
- Purchase cards provide a cleaner payment collection process for merchants. Merchants have the ability to collect their funds in conjunction with the settlement of their credit card transactions.

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 Card Data Levels

For a purchase card, three levels of data can be captured and sent by a merchant to the buyer organisation. They are:

Level I:

Level I transaction data consists of only basic data. A standard credit card transaction provides level I data to the processor. The buyer cannot derive any special benefits from purchase card usage if the merchant passes only level I data.

Level II:

Level II transaction data consists of data such as tax amount and order number in addition to level I data.

Level III:

Level III line item detail provides specific purchase information such as item description, quantity, unit of measure and price. This information is very useful to the buyer to help streamline accounting and business practices and to merge payment data with electronic procurement systems.

Note: Data in the table is only indicative. The actual fields are processor-dependent.

This table lists information on data that is passed by iPayment in each level.

Data	Level 1	Level 2	Level 3
Card Number	X	X	X
Card Holder Name	X	X	X
Card Expiration Date	X	X	X
Card Holder Billing Address	X	X	X
Currency Code	X	X	X
Tax Amount		X	X
Transaction/Order Number		X	X
Ship from Postal Code		X	X
Destination Postal Code		X	X
Discount Amount			X
Freight Amount			X
Duty Amount			X
Line Item Information			X

Processing Purchase Card Transactions

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

iPayment automatically recognizes purchase cards based on a set of seeded card number ranges. iPayment passes additional information to the payment system only during the close batch operation. Authorization and other settlement operations carry the same information for purchase cards as they do for credit cards.

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. The business flow is as follows:

- 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 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 a capture transaction to its processor.
- Funds are transferred from the issuing bank (customer’s bank) to the acquiring bank (merchant’s bank).
- Issues 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 appropriate projects, cost centers, general ledger, or purchase order accounts.

Understanding Inbound Bank Account Remittance

Oracle iPayment supports inbound bank account remittance for both business-to-consumer and business-to-business models. The inbound bank remittance functionality facilitates electronic transfer of payment amounts from a customer’s bank account to the payee’s bank account (pre-authorized direct debit). EC applications can use bank account transfers for Electronic Fund Transfers (EFT) transactions, also known as Automated Clearing House (ACH) transactions. Electronic Funds Transfer online validations are online and real-time while actual funds transfer transactions are offline. In addition to standard direct debits, Oracle iPayment supports remittance of bills receivable instruments. EC applications use Oracle iPayment as their interface to payment processors that provide connectivity to appropriate clearing house networks.

Note: Oracle iPayment has standard integration with Oracle Receivables for direct-debit and bills receivable remittance.

The number of operations supported for inbound EFT is less than 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 provided is debit instruction submission information. Oracle iPayment only supports offline payments for bank account transfers. See Understanding Offline and Online Payments for more information.

Electronic Funds Transfer (EFT) Online Validations

EFT online validations are a real time service provided by some payment systems to validate the bank account to be used in an EFT transaction. EFT online validations service ensures that the bank account instrument exists and there is no fraud alert. EFT funds transfer are not real time because one or two business days are needed to complete the transaction. Therefore, it is not possible to ensure that the bank account is still open and has sufficient funds. EFT online validation helps with a certain level of validity checking including:

- The validation step is an optional step for EFT transactions. The user can initiate it any number of times.
- The validation is performed in real time.
- The EFT online validation response message shares the same message standard with the credit card authorization response message for processor type payment systems.

Note: EFT online validation is only offered for United States ACH and not for all payment systems. EFT online validation checks whether a bank account exists and that the account is not flagged fradulent. EFT online validation does not reserve funds or check if the account has sufficient funds.

Interface with Electronic Commerce Applications

EC applications can use the same API for credit card, purchase card, and bank account payments. Oracle 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:

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

Note: Some credit card operations are not supported for bank account transfers. A payment system does not need to support all operations. Verify which operations are supported by your particular payment system.

Process Flow of an Inbound Bank Remittance Request

1. The EC application optionally calls the Oracle iPayment API to perform an online validation of the payer’s bank account.
2. The EC application calls the Oracle iPayment API to schedule an offline bank account transfer payment request.
3. 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.

4. 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.
5. A scheduled offline payment request can either be modified or canceled before it is routed to the payment system.
6. Once a request is routed to the payment system, the EC application can neither modify nor cancel the request.
7. The payment system routes the payment to the appropriate network.
8. If the payment processor is a payment gateway, then the EC application can query iPayment to retrieve the status of the payment transaction. iPayment, in turn, queries the payment system. If there is any failure in the payment network or at the payment system site while processing the payment, then the payment system responds with those errors.
9. Oracle iPayment updates its tables with the status of the transaction and returns to the EC application the status of the payment request.

Inbound Bank Remittance Requests Statuses

A bank account payment request in Oracle iPayment can have one of these statuses:

- **Pending**: After the EC 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 by the EC application.
- **Failed:** Failed due to technical errors.
- **Communication Error:** Failed due to communication errors. Transaction may be retried.
- **Unpaid:** Insufficient funds.

The status of a payment is determined by the status of the payment request. To obtain the status of a payment request, EC applications can call the Query Transaction Status API. See the Inbound Bank Remittance page in Oracle iPayment to obtain the status of a payment request.

Note: iPayment does not handle reconciliation of bank account remittance transactions.

Understanding PINless Debit Card Transactions

PINless debit card transactions is a type of payment method offered by some payment systems to selected industries that are traditionally viewed as "recurring billers". The consumer initiates the debit card payment process without providing the PIN. The merchant takes the full responsibility to authenticate the consumers and assumes 100% liability of the transaction and any subsequent adjustments.

The transaction is sent to the debit networks for processing. Currently, three debit networks: STAR, NYCE, and PULSE support PINless debit card payments.

Authorization and capture of PINless debit card transactions are handled in a single step by all payment systems.

After authorizing the PINless debit card transaction, no modification to the transaction can be made. After the payment request is approved, the consumer's account is debited in real time. Any dispute, error, or modification associated with the transaction can be handled offline between the consumer and the merchant.

The settlement step for PINless debit card transactions is flexible. While some payment systems, such as Paymentech, need this step to complete the transaction, some payment systems, such as Citibank for example, do not support this step. PINless debit card transactions, however, share the same batch with regular credit card and purchase card transaction. The payment system's responsibility is to differentiate the transactions and settle the transactions accordingly.

Process Flow for gateway-model payment system

Most gateway type payment systems handle PINless debit card transactions in a single step. After receiving the authorization request, the payment system will send the transaction to the debit network. The consumer's account (payer) is debited after the authorization request is approved. The merchant's account (payee) is credited sometime later. PINless debit card transactions are different from the process flow of credit card transactions where the authorization and fund capture are separated into two steps.

To implement the PINless debit card feature based on the current iPayment architecture, the authorization is broken into two separate steps in order to capture the PINless debit card request. These steps are performed by oraPmtReq and oraPmtCapture which are present in the iPayment API.

The authorization step sends the transaction to the payment system, saves the transaction into IBY schema, and returns the authorization response back to the calling product (similar to credit cards).

The EC Application (calling product) sends the transaction through the oraPmtReq API to the IBY schema that determines the payment system based on the transaction type and the routing rules. The iPayment servlet validates the request parameters. The iPayment Extract Engine calls the Extract and Delivery API and receives authorization response from the payment system. The Engine stores the capture record if it is authorized. The response is parsed from the payment system and is saved in the IBY schema. The results are sent back to the EC application.

During the capture step, the transaction is retrieved from iPayment schema. If the transaction is authorized, a "Capture Success" message is returned to the calling product.

The EC Application (calling product) sends the transaction through the oraPmtCapture API to the IBY schema that retrieves the transaction and routing information from the database. The iPayment servlet validates the request parameters. The iPayment Extract Engine validates the authorization result and assembles the response. The response is parsed from the payment system and is saved in the IBY schema. The results are sent back to the EC application.

Process flow for processor-model payment system

There is no difference in the process flow of authorization and capture steps between gateway-model payment systems and processor-model payment systems in PINless debit card. The authorization and capture are handled in a single request. Some processor type payment systems (such as Paymentech), an additional step settlement is required to complete the transaction. The merchant's account will be credited after settlement and the fund transfer is said to be completed.

The settlement in PINless debit card settlement is similar to credit card transactions. The settlement shares the same message standards, except for different records and fields populated for PINless debit cards.

The authorization sends the transaction to the payment system, saves the transaction into IBY schema, and returns the authorization response back to the calling product (similar to credit card).

The EC Application (calling product) sends the transaction through the oraPmtReq API to the IBY schema that determines the payment system based on the transaction type and the routing rules. The iPayment servlet validates the request parameters. The iPayment Extract Engine calls the Extract and Delivery API and receives authorization response from the payment system. The Engine stores the capture record if it is authorized. The response is parsed from the payment system and is saved in the IBY schema. The results are sent back to the EC application.

During the capture step, the transaction is retrieved from iPayment schema. If the transaction is authorized, a "Capture Success" message is returned to the calling product.

The EC Application (calling product) sends the transaction through the oraPmtReq API to the BatchCCPayment schema that determines the payment system according to the transaction type and routing rules. The iPayment Extract Engine calls the Extract and Delivery API and receives authorization response from the payment system. The response is parsed from the payment system and is saved in the BatchCCPayment schema. The results are sent back to the EC application.

During the settlement step, the batch is created and sent to the payment system.

The iPayment scheduler calls the batch close API. In the BatchCCPayment schema, the captured transactions are retrieved and sends the transactions to the iPayment Extract Engine. The iPayment Extract Engine calls the Extract and Delivery API and receives authorization response from the payment system. The response is parsed from the payment system and is saved in the BatchCCPayment schema. The results are sent back to the iPayment scheduler.

Note: iPayment presents PINless debit cards in a single step process. While iPayment’s backend functionality supports payment systems requiring one or two steps for debit cards, Electronic Commerce Applications initiate a single transaction.

Understanding Outbound Bank Account Payments

Oracle iPayment supports outbound bank account payment for both business-to-consumer and business-to-business models. The outbound bank account payment functionality facilitates electronic transfer of payment amounts to a supplier’s bank account from the merchant’s bank account. Oracle iPayment accepts payment instructions using different instrument types such as checks, wires, and EFT. After performing validations, Oracle iPayment forwards instructions to the appropriate back-end payment system. Each payment system routes the instructions appropriately depending on the country, network, and instrument.

Note: Oracle iPayment only supports payment instructions that originate from Oracle Payables. Payment instructions are only available with processor type payment systems.

Outbound EFT supports fewer operations than credit card payments due to current practices and processes involved in processing account transfers. Also, real time response for outbound bank account payments is not available due to the current practices in account transfer processing. The only status provided is for payment instruction submission to the processing network. Oracle iPayment only supports offline payments for bank account transfers. See Understanding Integration with Oracle Payables for more information.

Interface with Electronic Commerce Applications

Oracle iPayment only integrates with Oracle Payables for outbound bank account payments.

Oracle iPayment supports the following payment methods from Oracle Payables:

- Electronic Funds Transfer (ACH)
- Check
- Wire

Process Flow of a Outbound Bank Payment Request

These steps describe the process flow for a typical bank outbound payment instruction.

1. Build a payment batch in Payables

   Oracle Payables provides the payment-batch functionality to initiate and monitor the payment process for multiple invoices. Building payment batches involves selecting the invoices based on the selection criteria specified for the batch and building the payments for the selected invoices. Use the Standard Build Payments Program in Oracle Payables to implement the build functionality.

2. Format payment batch using Oracle iPayment Single Format Program in Oracle Payables

   Payment instructions move to Oracle iPayment when you format the payment batch with the single format program. You can now see payment transactions in the Oracle iPayment Outbound Payments page.

3. Confirm payment batch in Oracle Payables

   Oracle iPayment processes only confirmed transactions. You must confirm the payment batch in Oracle Payables before Oracle iPayment can process the transactions further.

4. Oracle iPayment scheduler picks up all payment transactions for a specific merchant in confirmed Oracle Payables payment batches that must be routed to the same payment system.

   A single file from Oracle iPayment to the payment system can contain payment instructions that belong to multiple Oracle Payables payment batches.

5. Oracle iPayment servlet validates all data based on the requirement specified by the back-end payment processors. All data that passes validation is appropriately formatted and forms part of the payment file that is sent to the bank. After successfully sending the file to the back-end payment processor, Oracle iPayment updates the status of the transactions.

   If Oracle iPayment fails to send the payment file to the processor, the transactions will have a Communication Error status.

Note: Oracle iPayment does not update any status in Oracle Payables or handle reconciliation and processing of bank account statements.

Outbound Bank Payment Request Statuses

An offline outbound bank account payment request in Oracle iPayment can be in one of these statuses:

- **Formatted**

  Oracle Payables has sent the batch to Oracle iPayment but the payment batch is not confirmed in Payables.

- **Open Batched**

The payment is part of confirmed payment batch in Oracle Payables.

- **Canceled**

  Oracle Payables has cancelled a payment batch after sending it to Oracle iPayment.

- **Failed**

  The payment failed due to validation errors at the servlet.

- **Communication Error**

  The payment failed due to communication errors between the servlet and the engine. You can retry the transaction using the Oracle iPayment scheduler task - EFTPBATCHRETRY.

- **Batch Pending**

  Oracle iPayment has submitted the payment transaction to the back-end system. Oracle iPayment updates the status after querying the payment system for the payment status.

- **BEP-error**

  The payment has failed at the payment system due a payment system-specific error. The payment system was unable to process the payment instruction. You must correct the data in Oracle Payables and resubmit.

- **Success**

  The payment system has successfully executed the payment instruction. The money has moved form the merchant organization funding account to the supplier/beneficiary account.

See the Outbound Payments page in Oracle iPayment to obtain the status of a payment request.

Note: You need to import the bank statements using some channel other than Oracle iPayment and reconcile the transactions.

Understanding Offline and Online Payments

Oracle iPayment supports two models of payment processing for credit/PINless/purchase cards:

- Online Payment Processing
- Offline Payment Processing

Inbound bank account remittance and outbound bank payment transactions are always offline.

The types of operations that you can process online depends on the back-end payment system type you have chosen: gateway or processor model and your operation model: host based or terminal based.

For processor model payment systems, authorization operations must be online and capture operations are offline. For gateway payment systems operated by a terminal-based merchant, both authorization and capture operations can be online.

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 EC application. Online transactions are supported for credit, purchase, and PINless debit cards. Online validation transactions are supported for Electronic Funds Transfer.

Offline Payment Processing

Offline payment processing is the model in which payment requests are not immediately forwarded to back-end payment processors. When an EC application makes a payment processing request in a scheduled mode, or if the payment is predated, the payment information is saved in the Oracle iPayment database and is sent to the payment processor at a later time.

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 EC applications.

States of Offline Credit Card Payment Requests

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

- **Pending**: After the EC application makes a request and before the scheduler routes the request to the payment system.
- **Canceled:** When a pending payment request is canceled.
- **Failed**: Failed due to technical errors.
- **Unpaid:** Insufficient funds.
- **Communication Error:** Failed due to communication errors. Transaction may be retried.
- **Success**: Transaction succeeded.

You can do follow-up operations (such as capture for an authorization, return for a capture) if the original transaction has a "Success" status.

How the Scheduling System Works

The Oracle iPayment scheduler provides the ability to handle payment transactions that cannot be processed in real time. These transactions may be of two kinds: transactions that can be processed after they are submitted to Oracle iPayment, or transactions where the back-end payment system cannot process requests in real time. Scheduling is also useful for automating recurrent tasks associated such as batch closes. Batch closes are performed in a processor-model payment system such as Paymentech.

You can configure the Oracle iPayment scheduler to perform specific tasks with each invocation. You can specify the tasks to be performed through task parameters. For more information on the task parameters, see Overview of Oracle iPayment APIs in the *Oracle iPayment Implementation Guide*. If no task parameters are given to the iPayment scheduler, then all tasks will be performed.

> **Note:** A task is performed only once every time the scheduler is invoked, even if the same task appears multiple times in the list of task parameters.
>
> Two instances of the scheduler must not be active at the same time, even if they are configured to perform different tasks.

This table shows Oracle iPayment scheduler task parameter names and task descriptions.

Task Parameter Name	Task Description
BATCHCLOSE	A credit card/purchase card batch close operation will be attempted for all payee accounts that are with processor-model payment systems (for example, Paymentech).
BATCHQUERY	A batch query will be attempted for all credit card/purchase card batches that have been successfully submitted to processor-model payment systems but have not received final transaction results yet.
BATCHRETRY	A batch close will be attempted for all processor-model batches that failed because of a communication error with the back-end payment system servlet. Since the back-end payment system may have potentially received and processed the batch the first time, a retry could lead to double-billing. You should do batch retries manually after the merchant confirms the loss of the first batch, rather than by the scheduler.

Task Parameter Name	Task Description
BANKACCOUNT	Offline bank account transactions are submitted to gateway-model payment systems.
CREDITCARD	Offline credit card transactions are submitted to gateway-model payment systems.
PURCHASECARD	Offline purchase card transactions are submitted to gateway-model payment systems.
PDCBATCHCLOSE	A PINless debit card batch close operation will be attempted for all payee accounts that are with processor-model payment systems.
PDCBATCHQUERY	A batch query is attempted for all PINless debit card batches that were successfully submitted to the processor-model payment systems but have not received final transaction results yet.
PDCBATCHRETRY	A batch close is attempted for all processor-model batches that failed because of a communication error with the backend payment system servlet. Since the back-end payment system may have potentially received and processed the batch the first time, a retry could lead to double-billing. You should manually do batch retries after the merchant confirms the loss of the first batch, rather than by the scheduler.
EFTBATCHCLOSE	A batch close is attempted for all *inbound bank remittance* payee accounts with processor-model payment systems.
EFTBATCHRETRY	A batch close will be attempted for all EFT processor-model batches that failed because of an error (please see below for error types) with the Back end payment system servlet. Since the Back end payment system may have received and processed the batch the first time, a retry could potentially lead to double-billing. It is recommended that batch retries be done manually after the merchant confirms the loss of the first batch, rather than by the scheduler. The three instances when a EFTBatchRetry could occur are: ■ A communication error occurs with the servlet or the processor. ■ The database fails (that is, DB shutdown) while the scheduler is running. ■ If the number of batches per day has exceeded the limit on that processor.
EFTPBATCHCLOSE	Payment engine attempts a batch close for all *outbound bank payment* payee accounts with processor-model payment systems.

Task Parameter Name	Task Description
EFTPBATCHRETRY	The scheduler attempts a batch close for all *outbound bank payment* processor-model batches that failed because of a communication error with the BEP servlet. The two instances when a EFTPBatchRetry could occur are: ■ A communication error occurs with the servlet or the processor. ■ Critical system error (that is, DB shutdown) while the scheduler is running.

Scheduling Concurrent Programs

To schedule concurrent requests:

1. Log on to Self Service Applications with a Payment Administrator responsibility.
2. Navigate to the Find Requests window.
3. Click Submit a New Request to open the Submit a New Request window.
4. Select the Single Request option.
5. In the Name field, select the iPayment Scheduler from the list of values.
6. Specify the list of tasks that the scheduler has to perform.
7. To define a schedule, click Schedule to open the Schedule window.

   Defining a schedule can be as simple as submitting as soon as possible or using a more complex schedule.
8. Click Submit.

Understanding Risk Management

Card transactions continue to grow in number, taking an ever-larger share of the payment system and leading to a higher rate of stolen account numbers and subsequent losses by banks and merchants. Improved fraud detection thus has become essential to maintain the viability of the payment system and merchants.

Banks have used early fraud warning systems for some years. iPayment provides similar risk management functionality for credit card and purchase card transactions for EC applications. Risk management functionality is provided 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 that handle amounts exceeding a specified amount.

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

Risk management functionality enables payees and EC 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 Oracle iPayment based on their business model. When a Payment Request API is called, the EC application can specify which formula to be used to verify the identity of their customers, assess their customer credit rating, and risk rating in a secure environment. When the EC application calls the Payment Request API with the risk formula specified, Oracle iPayment will evaluate the risk and in parallel send the authorization request to the payment system. After getting a response from the payment system, Oracle iPayment will return both the authorization code and the risk score to the EC application. The EC application has to now decide whether to continue with the transaction and make a payment capture or discontinue the transaction.

Alternatively, Risk API can be called independent of the Payment Request APIs. Using the Risk API separately allows merchants to evaluate risk first. Depending on the risk score, merchants may not want to send the payment request for authorization. This avoids the overhead of sending an authorization for a potentially risky transaction. Please note that when the EC application calls the Risk API separately, iPayment cannot evaluate the risk scores associated with AVS (Address Verification System). iPayment gets the AVS codes directly from the payment system during an authorization request. As no authorization request is sent in this scenario, iPayment cannot get AVS codes from the payment system and hence cannot evaluate risks scores associated with AVS.

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 a 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 that a payment request is made by the customer. 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 a 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 whether 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 set up 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 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 payer involved in a payment request. If a payer has a good history of payments over a long duration, then payments requested by this payer 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 or shipping address with the address that is maintained for the cardholder by the issuing bank. iPayment does address verification during an authorization request, by calling the payment system with the address and zipcode information along with the payment transaction information. The payment system then does the authorization and also returns various AVS codes to iPayment. 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 the United States of America.
- **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 setup 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 Oracle iPayment and the values of these risk factors are set up in Oracle Receivables. Oracle Receivables stores credit management information about customer accounts such as credit rating, risk rating, etc. The following are risk factors 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. Credit Rating 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 or 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 or services to the customer is risky.

iPayment Routing and Operation

iPayment accepts payment transactions from EC applications and routes them to the appropriate back-end payment systems. The customer uses a web browser or a client application to exchange data with a web or server-based EC application. The EC application sends payment requests to iPayment. Finally, iPayment routes the payment requests to the appropriate payment systems. Each Payee can have its own set of routing rules with its own set of priorities.

What Constitutes a Routing Rule?

Every routing rule is made up of three components -

- **Basic Rule Information** - This information is used to select and rank all the rules that may be applicable to a payment transaction. The basic rule information consists of Rule Name, Payee, Payment Instrument Type, Rule Priority and Status.
- **Destination Information** - The destination information specifies the back-end payment system to which the payment transaction should be routed. The destination information consists of Payment System and Payment System Identifier.
- **Routing Rule Conditions** - This specifies the conditions under which a rule becomes applicable to a payment transaction. A rule condition is comprised of a condition name, a criterion for the condition (such as -Amount, Currency, Organization ID, Card Type, Card Number and Bank Routing Number), the type of operation related to the criterion and the value of the criterion. Multiple rule conditions can be defined for a routing rule.

How Routing Works

Routing of a payment transaction is based on a set of routing rules set up in the iPayment user interface by the iPayment administrator. The routing engine finds the appropriate Payment System in the following sequence:

1. The routing engine retrieves the rules associated with the Payee and Instrument Type specified in the payment request.
2. The routing rule with the highest priority is evaluated first. If the values in the transaction match the conditions specified in the routing rule, the request is routed to the corresponding Payment System using the specified Payment System Identifier.
3. If the values in the request do not match the conditions specified, the routing rule with the next highest priority is evaluated.

4. In case the payment request values do not match any of the conditions specified, the transaction is routed to the default Payment System using the default Payment System Identifier.

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

iPayment supports credit cards, purchase cards and bank account transfers. The payment methods available depend 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 business 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 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.

Routing Rule Conditions

Routing rule conditions determine whether the rule is applicable to a payment transaction. A rule can have multiple rule conditions. A rule is applicable to a payment transaction only if the payment transaction can meet all the conditions for the rule. For example, a payee can route all Visa credit card transactions when the order amount is greater than 500 US dollars to Payment System C.

The following table lists the values in the Operation and Value fields for a selected payment instrument type and criterion.

Payment Instrument Type	Criterion	Operation	Value
Purchase Card	Org ID	Equal, Not Equal To	Specify a value.(Only digits allowed)
Purchase Card	Card Type	Equal, Not Equal To	Select a card type from the drop down list.
Purchase Card	Currency	Equal, Not Equal To	Select a currency from the drop down list.
Purchase Card	Amount	Greater than, Greater than or equal to, Less than, Less than or equal to	Specify a value.(Only digits allowed)
Purchase Card	Purchase Card Number	Equal, Not Equal To	* Specify a value.
Credit Card	Org ID	Equal, Not Equal To	Specify a value.(Only digits allowed)
Credit Card	Card Type	Equal, Not Equal To	Select a card type from the drop down list.
Credit Card	Currency	Equal, Not Equal To	Select a currency from the drop down list.
Credit Card	Amount	Greater than, Greater than or equal to, Less than, Less than or equal to	Specify a value.
Credit Card	Credit Card Number	Equal, Not Equal To	* Specify a value.

Payment Instrument Type	Criterion	Operation	Value
PINless debit Card	Org ID	Equal, Not Equal To	Specify a value. (Only digits are allowed)
PINless debit Card	Card Type	Equal, Not Equal To	Select a card type.
PINless debit Card	Currency	Equal, Not Equal To	Select a currency.
PINless debit Card	Amount	Greater than, Greater than or equal to, Less than, Less than or equal to	Specify a value.
PINless debit Card	Debit Card Number	Equal, Not Equal To	* Specify a value.
Bank Receipts-Direct Debit	Payer Bank Routing Number	Equal, Not Equal To	Specify a value.
Bank Account	Org ID	Equal, Not Equal To	Specify a value.
Bank Account	Currency	Equal, Not Equal To	Select a currency.
Bank Account	Amount	Greater than, Greater than or equal to, Less than, Less than or equal to	Specify a value.

* - Value can be digits, spaces, dashes and wild card character (%). For example, if the value is 4111%, then the routing rule applies to all card numbers that begin with "4111".

Understanding Transaction Reporting

Oracle provides management summaries directly from the transaction data. Transaction reporting (TR) users can view indicators on a daily, weekly, or monthly basis, targeted to their particular lines of business and summarized across all processors, types of cards, and transaction types.

Transaction reporting lets every manager in an organization of any size to know the state of business transacted on credit and purchase cards on a daily basis, and to make mid-course corrections that drive the business towards achieving its goals. Transaction reporting helps enterprises achieve consistent, high-integrity information, corporate wide alignment, and collaborative decision making. A proactive e-mail notification system relieves the burden of constantly monitoring critical measures. Through transaction reporting, Oracle iPayment provides an environment that supports mixed workloads, such as processing transactions versus running queries, without compromising on performance or scalability, providing the simplest, and therefore the least costly approach.

Functioning of the E-mail Push System

The Oracle iPayment e-mail push system provides the ability to send e-mail notifications to specified users which frees mail from the need to monitor critical measures.

Any user with the iPayment Daily Business Close User responsibility can run the e-mail push system by submitting a concurrent request. You can configure the Oracle iPayment e-mail push system to send e-mails on a pre-defined schedule. The reports provide a daily summary of transactions. For best results, schedule the process to run at the close of business everyday. Specify the recipients for the e-mail notification by providing the e-mail ID or user names as parameters for the concurrent task. The user names must have valid e-mail IDs associated with them. Specify multiple recipients for a notification by separating the e-mail IDs or user names by a comma (',').

The e-mail push system sends the following information to the receiver after summarizing the transactions for that day.

- Login URL
- Date that the report was generated
- Total number of transactions
- Total transaction amount
- Total number of authorization transactions
- Total authorization amount

- Total number of captured transactions
- Total captured amount
- Total number of credit/return transactions
- Total credit/return amount
- Total number of credit card transactions
- Total credit card transactions amount
- Total number of purchase card transactions
- Total purchase card transactions amount

Scheduling E-mail Push Programs

To schedule concurrent requests for e-mail push programs:

1. Log in to Self Service Applications as any user with the iPayment Daily Business Close User responsibility.

   Choose the iPayment Daily Business Close User if the user has multiple responsibilities linked to the user name.

2. Navigate to the Submit a New Request window.
3. Select the Single Request option.
4. From the Name choice list, select IBY Push E-mail Report.
5. Specify the recipient e-mail address. For multiple addresses, use the comma (’,’) as a separator.
6. To define a schedule, click Schedule.

   The Schedule window appears.

7. Define the schedule.

   Defining a schedule can be as simple as submitting as soon as possible or using a more complex schedule.

8. Click Submit**.**

Understanding iPayment Security

Oracle iPayment has several features to ensure the security and privacy of your data.

This section explains the security features of Oracle iPayment and describes the set up required to properly utilize these features.

Payment Engine

Oracle iPayment engine stores and processes highly sensitive financial data. To ensure proper security of this data, the Oracle iPayment engine has advanced security features.

Visibility Restriction in Operations UI

The Oracle iPayment engine has advanced security features to ensure that unauthorized personnel cannot view sensitive data in the Oracle iPayment schema. Oracle iPayment masks out sensitive information according to the visibility classes in the pages. All sensitive data are masked before writing into the various log files. For more information on visibility restrictions in the Operations pages, see Understanding Visibility Class.

Encryption Using Payee Security Key

For each payee configured in Oracle iPayment, you can define a security key. The security key is specific to a payee and is not stored in the schema. Whenever you bounce the web server running Oracle iPayment, you should log in to the Oracle iPayment security page and re-enter your payee key. The security key serves several purposes:

- The payment engine sends the security key to the servlet during the EFTPBATCHCLOSE/EFTPBATCHRETRY and EFTBATCHCLOSE/EFTBATCHRETRY operation. The servlet uses the security key to open your certificate and retrieve your private key. You must also appropriately configure the servlet to use the payee key by setting the “CRYPTOGRAPHIC DISCIPLINE”option.
- The security key provides data privacy. Sensitive data such as credit card, purchase card, and bank account numbers are encrypted in the database. For security enabled payees, sensitive data in a transaction is encrypted using the payee key.
- To enable encryption of all registered instruments for all payees, specify the "Instrument Registration Key" in the Security page.

iPayment Engine to iPayment Servlet Communication

Oracle iPayment architecture lets you install the servlet in a machine outside the firewall. If you have installed either Oracle iPayment (or its components) or the EC application in a distributed environment, Oracle recommends configuring SSL between Oracle iPayment and the payment system components. You can create an Oracle Wallet to store certificates and credential information to support authentication of the engine, in this case a client of the servlet, by the server running the servlet. You can specify the wallet location and password using FND profiles. You can configure the server where the servlet is running to request for client certificates (on engine side) and Oracle iPayment retrieves the certificates from the Oracle Wallet and sends the certificates to the server for authentication. Oracle iPayment also supports basic authentication of the payment system by the servlet.

These security features are recommended to guard against unauthorized access to data and Oracle iPayment services. Oracle iAS web server (Apache Server) provides several types of authentication that you can use to secure servers, listeners, and servlets.

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 Oracle iPayment:

- Install all the following components on the same machine:
  - iPayment
  - Payment system servlet
  - EC application
- Use Secure Socket Layer (SSL) to connect distributed components

Secure Socket Layer

If either Oracle iPayment (or its components) or the EC application is installed in a distributed environment, Oracle recommends enabling SSL communication between Oracle iPayment and the payment system components.

Basic Authentication for Payment Systems

For setting up security for basic authentication, you must perform some tasks both in Oracle iPayment administration user interface and in Apache Server administration tool. While configuring Oracle iPayment for a particular payment system using the Oracle iPayment administration user interface, you must assign the payment system user name and password in the payment system configuration screens. You must assign the same user name and password in the Apache Server that runs the payment system servlet(s).

For details on setting up basic authentication in Apache Server, see the Apache Server documentation.

IP Address Restriction

In addition to using the merchant user name 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 from which the web server should accept transaction requests for iPayment)
- The IP addresses of some non-trusted hosts (machines from which the web server should refuse transaction requests for iPayment)

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.

Other Security Related Information

- EC applications can be built to use iPayment's Java APIs. Since this approach avoids the EC App servlet, it prevents the network transfer of sensitive information between EC applications and iPayment.
- Separate HTTP ports for site administration and iPayment administration increases security.
- Security can be increased by using SSL for communication between iPayment and the payment system servlet.

Understanding Visibility Class

Visibility classes are used to determine what data is visible in Oracle iPayment Operations UI. Visibility classes are created and updated by any user with iPayment System Administrator responsibility and are assigned to users and responsibilities through iPayment profile options.

What Constitutes a Visibility Class?

Every visibility class is made up of three components:

- **Basic Visibility Class Information** - The visibility class information is used to identify a visibility class. The basic rule information consists of visibility class name, effective from date, and effective to date.
- **Data Masking Information** - The data masking information specifies how the instrument number is displayed in the Operations page, such as displaying the amount and the beneficiary information.

This table shows details on the masking information fields.

Attribute	Allowed Values	Description
Instrument Number Mask	Show All	Full number is displayed on the pages.
Instrument Number Mask	Show First Four digits	The first four digits of an account number or credit card number associated with a transaction is displayed on the pages. Rest of the digits are masked.
Instrument Number Mask	Show Last Four digits	The last four digits of an account number or credit card number associated with a transaction is displayed on the pages. Rest of the digits are masked.
Instrument Number Mask	Show None	Number is fully masked. No digit is displayed.
Amount Mask	Yes	Amount is masked on the pages.

Attribute	Allowed Values	Description
Amount Mask	No	Amount is not masked on the pages.
Beneficiary Information Mask	Yes	No information regarding beneficiary is displayed.
Beneficiary Information Mask	No	All information regarding the beneficiary is displayed.

- **Visibility Conditions** - Visibility conditions specify the conditions under which transactions are displayed to a user in this visibility class. A visibility condition is comprised of a payee, organization (operating unit), and the e-commerce application. You can define multiple visibility conditions for a visibility class. Only transactions that meet the conditions are displayed on the pages.

> **Note:** A visibility class must have at least one visibility condition before it can be assigned to users for viewing Oracle iPayment transactions.

This table shows details on the visibility condition attributes.

Attribute	Description
Payee	You can select "All Payees" or select specific payees. This is useful in situations where a single iPayment talks to multiple payees. By defining separate visibility classes for each payee, and assigning it to the profile option at user level, you can ensure that employees of one payee do not have access to data of another payee.
Organization	You can select "All Organizations" or a specific operating unit. This allows you to filter data based on the Operating Unit that originates the transaction. This is useful in situations where a single iPayment talks to multiple Payables or Receivables operating units. By defining separate visibility classes for each organization unit, and assigning it to the profile option at site level, you can ensure that employees of one operating unit do not have access to data of another operating unit.

Attribute	Description
e-Commerce Application	You can select "All e-Commerce Applications" or specific application. This allows you to filter data based on the application that originates the transaction. This is useful in situations where a single iPayment talks to multiple applications. By defining separate visibility classes for each application, and assigning it to the profile option at responsibility level, you can ensure that a payables clerk does not see data originating from payroll.

iPayment has the following seeded visibility classes:

Default Payment Administrator Visibility

This table shows the Payment Administrator visibility class attributes:

Attribute	Value
Instrument Number Mask	Show Last Four Digits
Amount Mask	Yes
Beneficiary Information Mask	No
Payee	All Payees
Organization	All Organizations
e-Commerce Application	All e-Commerce Applications

Payroll Clerk Visibility

This table shows the Payroll Clerk visibility class attributes:

Attribute	Value
Instrument Number Mask	Show Last Four Digits
Amount Mask	Yes
Beneficiary Information Mask	Yes
Payee	All Payees
Organization	All Organizations

Attribute	Value
e-Commerce Application	Oracle Payroll

Receivables Clerk Visibility

This table shows the Receivables Clerk visibility class attributes:

Attribute	Value
Instrument Number Mask	Show Last Four Digits
Amount Mask	No
Beneficiary Information Mask	No
Payee	All Payees
Organization	All Organizations
e-Commerce Application	Oracle Receivables, iPayment, Oracle Order Entry, Oracle Order Capture

How Visibility Class Works

Data visibility on the Operations pages is based on the visibility class linked to the Oracle iPayment profile option associated with a user. Users with System Administrator responsibility assign a visibility class to the profile option. Visibility classes are created in Oracle iPayment by users with the iPayment System Administrator responsibility. The division of responsibility ensures that a user who creates a visibility class does not have the privilege to assign the visibility classes to other users or to them self.

When a user logs in and accesses the Operations pages, the Oracle iPayment engine retrieves the value of the IBY: UI Visibility Class profile option. To retrieve this value, Oracle iPayment searches through the user level, responsibility level, application level, and finally the site level for the profile value.

The IBY:UI Visibility Class profile value is seeded at the site level to the visibility class Default Payment Administrator Visibility. If you do not define and assign any new visibility class, all users will have the privileges associated with this seeded visibility class.

Depending on the visibility class, Oracle iPayment retrieves the transactions that meet the visibility conditions, and displays the transactions on the pages as specified by the masking set-up.

Oracle iPayment supports credit cards, purchase cards, and bank account transfers. The payment methods available depend on the payment system that you decide to use.

Payees and businesses can customize how Oracle iPayment displays transactions in the Operations pages using visibility classes based on their business requirements. For example:

- A business wants only select users to view payroll data.
- A business wants certain users to have access to all data.
- A business wants certain users to see only certain aspects of the transaction data.
- A business wants to filter data access based on a combination on payees, operating units and originating applications.

Understanding Extensibility

Extensibility allows interaction between iPayment and a back-end payment system to be customized. Note that extensibility only exists for Gateway-model payment systems.

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

Extensibility typically has the following workflow:

1. The EC application integrating with Oracle iPayment first writes custom back-end payment system parameters to the database.
2. It then sends a transaction request to iPayment, during which the extensibility class that was implemented queries the custom parameters and adds them to the request.
3. After a back-end payment system response is generated, the extensibility class is called again and custom parameters sent by the back-end payment system into the database are written. These parameters are queried later by the EC application or the extensibility class itself, which can use them for follow-on transactions.

   The extension elements that are returned in the response to the authorization API. The elements are available in the extract for capture transactions during batch close. A table stores the extensible parameters populated by the names/values that are returned by the system profile’s acknowledgement parser.

> **Note:** You do not need to use extensibility when integrating with back-end payment systems for which Oracle iPayment provides out-of-the-box servlets. If you add custom extensibility to such servlets, you might need to re-certify the system again.

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 set up and administrative functions of Oracle iPayment are done through the Oracle iPayment user interface. You can log in, create and modify payment systems, payees, risk management properties, and routing rules. You can also deactivate payees and routing rules.

Oracle iPayment is administered through a browser-based administration user interface that is implemented using the Oracle Self-Service framework. Administering iPayment includes using the Oracle iPayment user interface to configure Oracle iPayment security features, to add and configure the payment systems, payees, routing rules, and risk management. You can also use Oracle iPayment pages to query the transaction statuses. You can also test credit card operations online.

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.

Tab Name	Functionality
Security	Enable and modify iPayment security features
Setup	Create and configure the payment systems, payees, routing rules, and risk management.
Operation	View operations for all payment instrument types. You can also initiate test online credit card operations using these screens.
Visibility	Create, modify and de-activate visibility classes.

Navigating the iPayment Administration User Interface

The iPayment administration user interface includes the administration tabs, their related subtabs, and the administration workspace.

The administration tabs are on the top of the page, and remain visible as you navigate through Oracle iPayment. The tabs list the administrative tasks that you can perform. When you click a tab, you see a horizontal navigation bar with the subtabs associated with the selected tab. When you click a subtab, details for the selected task appear in the administration workspace in the lower portion of the page.

Managing Security

A user interface is provided in Oracle iPayment to manage security features of Oracle iPayment. Access the interface by clicking the Security tab. This is the default tab that is displayed when you navigate into the Oracle iPayment pages.

Through the user interface, you can:

- Create and modify security keys for active payees and instrument registration.
- Enter security key for the current server session.
- Enable and disable security for payees.
- Enable and disable security for instrument registration.

For more details on the security features of Oracle iPayment, see Understanding iPayment Security.

> **Note:** Remember to log into the Oracle iPayment security pages and re-enter the payee key whenever you bounce the web server running Oracle iPayment. Otherwise, the Oracle iPayment engine cannot submit transactions involving security-enabled payees or secure registered instruments. For load balancing using multiple JVMs, please ensure that you enter the key for each JVM.

The following table lists the subtab names and the functionality available from the security user interface.

Tab Name	**Functionality**
Session Security	View the security setups and enter security key for the current server session.
Security Management	Enable or disable security and manage the security keys for Registered Instruments and for Payees.

Session Security

The Session Security page lists all the active payees, total active payees with security enabled, and total security enabled payees for the current session.You can only view the security setup for the current session on this page.

- Enter Security Key for Payee for Current Session
- Enter Security Key for Instrument Registration for Current Session

Enter Security Key for Payee for Current Session

Once you enable security for an active payee, the key is stored in memory. Therefore, every time you bounce the server which hosts Oracle iPayment, you need to enter the key once again using the Session Security page.

Prerequisites

You must enable security for the payee and create a key.

To enter security key for a payee:

1. Navigate to the Session Security page.
2. Enter a security key for the Payee.
3. Click Apply.

Enter Security Key for Instrument Registration for Current Session

Once you enable security for instrument registration, the key is stored in memory. Therefore, every time you bounce the server which hosts Oracle iPayment, you need to enter the key once again using the Session Security page.

Prerequisites

You must enable security for instrument registration and create a key.

To enter security key for instrument registration:

1. Navigate to the Session Security page.
2. Enter a security key for the Instrument Registration.
3. Click Apply.

Security Management

Use the Security Management page to enable or disable security and manage the security key for registered instruments and for active payees.

- Enable Security for Payee/Instrument Registration
- Disable Security for Payee/Instrument Registration
- Modify Security Key for Payee/Instrument Registration

Enable Security for Payee/Instrument Registration

To enable security and create a key for active payees or instrument registration:

1. Navigate to the Security Management page.
2. Only active payees are shown in this page.

   If no key has been created for a payee, only the link for creating a key is enabled.
3. Select Enable from the Enable Security choice list for the payee (or instrument registration) that you want to enable security for.
4. Click on the Create Key icon for the payee that you want to create a security key for.

   This opens the Create Payee Security Key page.
5. Enter key and reconfirm key.

   Security keys must be composed of more than 6 alpha numeric characters. Leading and ending spaces are ignored.
6. Click Apply.

   A new security key is created for the payee and security for the current session is enabled for the payee.

Disable Security for Payee/Instrument Registration

To disable security for active payees or instrument registration:

1. Navigate to the Security Management page.
2. Select Disable from the Enable Security choice list for the payee (or instrument registration) that you want to disable security for.
3. Click Apply.

   Security for the payee or instrument region is now disabled.

Modify Security Key for Payee/Instrument Registration

Prerequisites

You have already created a security key.

To modify a key for active payees or instrument registration:

1. Navigate to the Security Management page.
2. Only active payees are shown on this page.

   If a key exists, only the link for changing the key is enabled.
3. Click on Change Key icon for the payee that you want to modify the security key for.

   This opens the Change Payee Security Key page.
4. Enter information for the old key, new key, and reconfirm new key.

   Security keys must be composed of at least 6 alpha numeric characters. Leading and ending spaces are ignored
5. Click Apply.

iPayment Setup

A user interface is provided in Oracle iPayment to let you perform various setup tasks. Access this by choosing the Setup tab. The Setup tab allows you to create and configure the payment systems, payees, routing rules, and risk management.

The following table lists the subtab names and the functionality available from the setup tab user interface.

Tab Name	Functionality
Payment System	Create and modify payment system properties in iPayment.
Payee	Create and modify payee properties and risk management properties in iPayment.
Routing Rule	Create and modify routing rules in iPayment.
Operation	Search and view iPayment transactions. Also perform tests for online credit card operations.

Visibility Class

The Visibility Class page lists all the visibility classes, data masking associated with the visibility class, and links to update the visibility class details.

- Viewing Visibility Classes
- Creating a Visibility Class
- Modifying a Visibility Class
- De-activating a Visibility Class

For more information, see Understanding Visibility Class.

Viewing Visibility Classes

To view visibility classes:

- Login as a user with the iPayment System Administrator responsibility. Click the Visibility Class tab.

  The Visibility Class page appears.

For more information, see Understanding Visibility Class.

Creating a Visibility Class

To create a visibility class:

1. Login as a user with the iPayment System Administrator responsibility. Click the Visibility Class tab.

   The Visibility Class page appears.

2. Click Create Visibility Class**.**

   The Visibility Class Details page appears.

3. Enter values in the fields on this page.

4. Click Apply.

   A new visibility class is created. You must add visibility conditions to this class before you can assign the class to users to control data visibility.

5. Define visibility conditions for the visibility class. Select the payee, organization, and e-Commerce application to which this visibility class applies. Click Add Another Row button.

   The new visibility condition is added to the visibility class.

6. Click Apply.

   A new visibility condition is added to the class.

Modifying a Visibility Class

To modify a visibility class:

1. Login as a user with the iPayment System Administrator responsibility. Click the Visibility Class tab.

   The Visibility Class page appears.

2. Click Update icon against the Visibility Class you need to modify**.**

   The Visibility Class Details page appears.

3. Modify the fields.

4. To remove a visibility condition, select one or more conditions and click on Delete button.

5. Click Apply.

For more information, see Understanding Visibility Class.

De-activating a Visibility Class

To inactivate visibility class rules:

1. Login as a user with the iPayment System Administrator responsibility. Click the Visibility Class tab.

   The Visibility Class page appears.

2. Click Update icon against the Visibility Class you need to de-activate**.**

   The Visibility Class Details page appears.

3. Set the ’Effective to’ date to the date from which you want the Visibility Class to be inactive.

4. Click Apply.

For more information, see Understanding Visibility Class.

Payment System

The Payment Systems page lists all the registered payment systems, whether the payment system is the default payment system for different payment instrument types, and links to update the payment system details.

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

Creating a New Payment System

To create and register a new payment system:

1. Navigate to the Payment Systems page.
2. Click Create Payment System**.**

   The Create Payment System page appears.
3. Enter payment system details.
4. Click Apply.

   The payment system is permanently registered.

Modifying a Payment System

To modify the values associated with a payment system:

1. Navigate to the Payment Systems page.
2. Click the update icon for the payment system name that you want to modify.

   The Update Payment System page appears.
3. Modify the fields associated with a payment system.
4. Click Apply.

Updating a Default Payment System

To update the default payment system for an instrument type:

1. Navigate to the Payment Systems page.
2. Select a payment system by clicking on the radio button next to the payment system name.
3. For the selected payment system, choose a payment instrument from the Set as Default choice list on top of the payment system table.
4. Click Go.

   Any transaction not routed by the routing rules is routed to the default payment system based on its instrument type. The selected payment system is set as default for the chosen payment instrument type.

Payees

The Payees page displays a list of all the registered payees, their status, and their associated risk management links.

- Creating a New Payee
- Modifying a Payee
- Inactivating a Payee
- Enabling Risk Management for Payee

Creating a New Payee

To create a new payee:

1. Navigate to the Payees page.
2. Click Create Payee.

   The Create Payee page appears.
3. Add payee details.
4. Select the payment instrument and enter the merchant category code.
5. Click Apply.

   A new payee is created. You can now add payment system merchant parameters and payee account information for the payee. When a payee is created, Oracle iPayment also creates a default risk formula that is associated with the payee.
6. To add Identifiers for the payment systems, click the Update icon next to the appropriate payment system name.

   The Update Payee Payment System Identifiers page appears. On this page you can add payment system identifiers for a payee/payment system combination and set one of the identifiers as the default for payment routing. While routing a transaction, if none of the rules match the transaction values, the default payment system identifier for the default payment system is used. On this page, you can also complete the payer account information for a payment system identifier. Click the Enter Parameters icon next to the payment system identifier. In the Payee Account Information page fill out the account and connectivity parameters that the payment system uses to process payment requests from this payee account.
7. Click Add Another Row to add identifiers.

   Enter the identifiers and select one as the default identifier.
8. Click Apply.

   The Payee Details page appears again.

Modifying a Payee

To change a payee's properties:

1. Navigate to the Payees page.
2. Click the Update Payee icon for the payee that you want to modify.

   The Update Payee page appears.
3. Modify the fields associated with a payee.
4. Click Apply.

   The Payee information is updated. You can now add payment system merchant parameters and payee account information for the payee.
5. To modify merchant identifiers for a payment system, click the Update icon next to the payment system that you want to add or modify the merchant identifier for.

   The Update Payee Payment System Identifiers page appears. On this page you can add payment system identifiers for a payee/payment system combination and set one of the identifiers as the default for payment routing. While routing a transaction, if none of the rules match the transaction values, the default payment system identifier for the default payment system is used. On this page, you can also complete the payer account information for a payment system identifier. To do so, click the Enter Parameters icon next to the payment system identifier. In the Payee Account Information page fill out the account and connectivity parameters that the payment system uses to process payment requests from this payee account.
6. Click Add Another Row to add identifiers.

   Enter the identifiers and select one as the default identifier.
7. Click Apply.

   The Payee Details page appears again.
8. Click Apply.

Inactivating a Payee

To inactivate a payee:

1. Navigate to the Payees page.
2. Click the Update Payee icon for the payee that you want to modify.

   The Update Payee page appears.
3. Choose Inactive from the Status choice list.
4. Click Apply.

Enabling Risk Management for Payee

To update the risk management status of each payee:

1. Navigate to the Payees page.
2. Click the Update Payee icon for the payee that you want to modify.

   The Update Payee page appears.
3. Choose Enabled from the Risk choice list.
4. Click Apply.

For more information, see Understanding Risk Management.

Risk Factors

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

For more information, see Understanding Risk Management.

Modifying the Risk Score

To modify the risk score:

1. Navigate to the Payees page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Risk Score from the Risk Factors choice list.

   The risk scores details appear.
4. Enter the risk score associated with each risk level.
5. Click Apply.

Modifying the Payment Amount Limit Risk Factor

Payment amount is the amount involved in a payment request. The payment amount scale varies from business to business. Based on the business model, each risk level varies with different amount ranges.

To modify the payment amount limit risk factor:

1. Navigate to the Payees page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Payment Amount Limit from the Risk Factor choice list.

   The payment amount limit details appear.
4. For each risk level, enter a positive integer representing the lower bound of the amount range in payment amount limit column. For example, if you set the amount against medium to 1000 and medium-high to 2000, payments equal to or greater than 1000 but less than 2000 are classified as medium risk.
5. Click Apply.

Modifying the Payment History Risk Factor

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

To modify the payment history risk factor by configuring the frequency values:

1. Navigate to the Payees page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Payment History from the Risk Factor choice list.

   The payment history details appear.
4. Enter a positive integer for the duration value and select a duration period.
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 Apply.

Modifying the Transaction Amount Risk Factor

The 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.

To modify the transaction amount risk factor:

1. Navigate to the Payees page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Transaction Amount from the Risk Factor choice list.

   The transaction amount details appear.
4. Enter positive numbers for payment and duration values, and select a duration period.
5. Click Apply.

Modifying the Time of Purchase Risk Factor

The time of purchase is the time that a payment request is made by the payee’s customer. A risk level can be associated with every hour of the day. No hour can be associated with more than one risk level.

To modify the time of purchase risk factor:

1. Navigate to the Payees page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Time of Purchase from the Risk Factors choice list.

   The time of purchase details appear.
4. For each time range, select the starting and ending hour and its risk level.
5. Click Apply.

You can add more time ranges and their risk levels by entering time ranges and risk levels in the last row of the table and clicking on Add Another Row button. You can also delete time ranges by clicking the corresponding trash can icon in the Delete column. No time ranges should overlap.

Modifying the AVS Codes Risk Factor

Address Verification Service (AVS) 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.

To modify the AVS codes risk factor:

1. Navigate to the Payees page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Address Verification Codes from the Risk Factor choice list.

   The AVS codes details appear.
4. Enter the AVS codes (separated by commas) corresponding to each risk level.
5. Click Apply.

> **Note:** If you remove all existing AVS codes, Oracle iPayment restores the default values.

Modifying the Frequency of Purchase Risk Factor

The Frequency of Purchase risk factor is used to track sudden surges 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, the payment request is considered to be a high risk payment.

To modify the frequency of purchase risk factor:

1. Navigate to the Payee page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Frequency of Purchase from the Risk factors choice list.

   The frequency of purchase details appear.
4. Enter positive numbers for payment and duration values and select a duration period.
5. Click Apply.

Modifying Oracle Receivables Risk Codes Risk Factor

Use this procedure to modify Oracle Receivables risk codes risk factor from the Risk Factors page. 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.

To modify Oracle Receivables risk codes risk factors:

1. Navigate to the Payees page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Oracle Receivables Risk Codes from the Risk Factors choice list.

   The Oracle Receivables risk codes details appear.
4. Select risk levels corresponding to Oracle Receivables risk codes in each row.
5. Click Apply.

Modifying Oracle Receivables Credit Rating Codes Risk Factor

Credit rating is the information enabling payees to effectively manage financial terms with their customers and is useful for online financing or for evaluating purchases of large amounts for a new customer. You associate risk levels to each credit rating for a payee.

Prerequisites

Install and register Oracle Receivables.

To modify Oracle Receivables credit rating codes risk factors:

1. Navigate to the Payees page.
2. Click the Update Risk Factors icon for the payee that you want to update.

   The Setup Risk Factors page for the payee appears.
3. Select Oracle Receivables Credit Rating Codes from the Risk Factors list.

   The Oracle Receivables credit rating codes details appear.
4. Select the risk levels corresponding to Oracle Receivables credit rating codes in each row.
5. Click Apply.

Modifying the Risky Instruments Risk Factor

The risky instruments 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 charge backs. 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

The Ship To/Bill to Address 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 Oracle Receivables Transactional Credit Limit Risk Factor

The Oracle Receivables Transactional Credit Limit 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 Oracle Receivables Overall Credit Limit Risk Factor

The Oracle Receivables Overall Credit Limit 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 page. The Risk Formula page lists all the risk formulas available for a 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.

For more information, see Understanding Risk Management.

Creating a Risk Formula

To create a risk formula:

1. Navigate to the Payees page.
2. Click the Update Risk Formulas icon for the payee that you want to update.

   The Risk Formulas page appears. The Risk Formulas page lists the implicit formula and other risk formulas for the payee.
3. Click Create Risk Formula.

   The Update Risk Formula page 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 page.
6. Click Apply.

For more information, see Understanding Risk Management.

Updating a Risk Formula

To update a risk formula:

1. Navigate to the Payees page.
2. Click the Update Risk Formula icon for the payee that you want to update.

   The Risk Formulas page appears. The Risk Formulas page lists the implicit formula and other risk formulas for the payee.
3. Click the Update icon for the risk formula to be modified.

   The Update Risk Formula page 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 page.
5. Click Apply.

For more information, see Understanding Risk Management.

Deleting a Risk Formula

To delete the risk formula, except the implicit risk formula:

1. Navigate to the Payees page.
2. Click the Update Risk Formula icon for the payee that you want to update.

   The Risk Formulas page appears. The Risk Formulas page lists the implicit formula and other risk formulas for the payee.
3. Click the Delete icon for the risk formula that you want to delete.
4. Click Done.

Routing Rules

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

- Viewing Routing Rules
- Creating Routing Rules
- Modifying Routing Rules
- Inactivating Routing Rules

For more information, see iPayment Routing and Operation.

Viewing Routing Rules

To view routing rules:

1. Navigate to the Routing Rules page.
2. Use the search criteria to narrow the list of routing rules. Set the required filter conditions and click Search.

   By default, iPayment displays the routing rules for all payees.

For more information, see iPayment Routing and Operation.

Creating Routing Rules

To create a routing rule:

1. Navigate to the Routing Rules page.
2. Click Create Routing Rule**.**

   The Create Routing Rule page appears.
3. Enter a rule name.
4. Verify that the status of the rule is set to Active. Inactive rules will not be applied.
5. Select a payment instrument from the choice list.
6. Select a payee, payment system, and payment system identifier.
7. Click Apply.
8. Enter a routing rule priority.
9. Define rule conditions for the rule. Select the criterion and click Add New Condition button.

   The new rule condition is added to the rule.
10. Enter a condition name, applicable operation, and value.
11. Click Apply.

    A new routing rule is created.

Modifying Routing Rules

To modify routing rules:

1. Navigate to the Routing Rules page.
2. Click the Update icon for the routing rule name that you want to modify.

   The Update Routing Rule page appears with all the routing rule fields.
3. Modify the fields.

   You cannot modify the Payment Instrument Type and Payee fields.
4. Click Apply.

For more information, see iPayment Routing and Operation.

Inactivating Routing Rules

To inactivate routing rules:

1. Navigate to the Routing Rules page.
2. Click the Update icon for the routing rule name that you want to modify.

   The Routing Rule Details page appears with all the routing rule fields.
3. Select Inactive from the Status choice list.
4. Click Apply.

Managing Operations

A user interface is provided in Oracle iPayment to test authorizations and capture operations for online processing of credit cards, PINless debit cards and purchase cards, view inbound bank remittance and outbound bank payments. You can also use the user interface to search and view details on actual transactions that were submitted through Oracle iPayment.

Note: PINless debit card transactions do not appear in the Operations user interface.

For credit and purchase cards, you can request a test authorization request can be submitted through the user interface 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 Oracle iPayment software to test iPayment setup. It should not be used for processing real payments.

The user interface can also be used to view and search on all transactions that were processed by Oracle iPayment. You can search transactions using various search criteria. Transactions matching the criteria are displayed in a summary format. You can view more details on a transaction can be viewed by selecting a transaction from the summary list. The Oracle iPayment user interface also provides the ability to submit follow-up operations for credit/purchase cards from the search pages.

Performing a Test Card Payment Authorization

Use these steps to test if the payment system, payee, risk factors, risk formulas, and routing rules are all set up correctly within Oracle iPayment.

Prerequisites

- Set up the Payment System, Payee, and Routing Rules during iPayment installation.
- Set up the Risk Factors and Risk Formulas, if you are also testing Risk Analysis.

To perform a test card authorization:

1. Navigate to the Search Card Operations page.
2. Click New Operation.

   This opens the Step 1:Transaction Basics page. This page is the first of a set of three steps for online Credit Card and Purchase Card Authorization/Authorization and Capture operations.
3. Enter the transaction basics and click Next**.**

   This opens the Step 2: Transaction Details page. This page is the second of a set of three steps for online Credit Card and Purchase Card Authorization/Authorization and Capture operations.
4. Enter the transaction details and click Next**.**

   This opens the Step 3: Review and Submit page. This page is the last of a set of three steps for online Credit Card and Purchase Card Authorization/Authorization and Capture operations.
5. Click Submit if you are satisfied with the information entered.

   The Authorization Results page appears with the entire response from the system about the Authorization success or Authorization failure.
6. Click Done to complete the Authorization operation.
7. To make changes to the selections made in case of a failed operation, click Back to navigate back through the steps.

Searching for Credit Card Transactions

You can search for transactions based on one or more criteria in the Main Card Transactions page. 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 search and drill down to view the details for all credit card and purchase card transactions from the Main Card Transactions page. Also, depending on the status of the transaction, you can initiate follow-on transactions such as capture or re-authorization for failed authorization operation.

To search for credit card transactions:

1. Navigate to the Credit Card and Purchase Card page.
2. Enter values for the search criteria.
3. Select all relevant transaction types and status from the check boxes.

   If you do not specify any criteria, the most recent 250 credit card/purchase card operations are displayed in the result table.
4. Click Search.

   The Search Results table appears with the details, based on the search criteria entered.

After performing a search, the Search Results table displays the summary information for transactions matching the search criteria.

> **Note:** Transactions can be sorted either by Ascending/Descending Date or Order/Tangible ID. Default sorting is done by TangibleID field first and then by Last Update Date.

Performing a Capture Operation

You can perform a capture operation starting at the Transaction Search Results table.

Prerequisites

- Identify the transaction for which the capture operation is to be performed.
- The transaction supports capture as a follow-on operation.

To perform a capture operation:

1. Click Capture in the Follow-on Operations column on the Transaction Search Results table.

   The Perform Capture page appears. By default, the Amount to Capture field contains the authorized amount.

2. If you want to perform the capture operation for an amount different from the authorized amount, edit the Amount to Capture field and click Submit.

   The Capture Results page 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 Credit Card Transaction Details

To view the details of a transaction from the Transaction Search Results table:

- Click the icon in the Details column for the transaction that you want to view the details for.

  The Credit Card Transaction Details page appears.

Searching for Inbound Bank Remittance Transactions

To search for inbound bank remittance transactions:

1. Navigate to the Inbound Bank Remittances page.
2. Enter values for the search criteria. To narrow the search criteria, you must enter as many search values as possible. You must specify a date in the Transaction Request Start Date field for the search.
3. Click Search.

   After performing a search, the Search Results table displays the summary information for transactions matching the search criteria. You can drill down to view the details for inbound bank remittance transactions from the table.

   > **Note:** Transactions can be sorted by all fields in the result table. Default sorting is done by transaction reference first and then by request date.

- Viewing Inbound Bank Remittance Transaction Details
- Viewing Inbound Bank Remittance Message Details

Viewing Inbound Bank Remittance Transaction Details

To view the transaction details:

- Click the icon in the Details column corresponding to the transaction that you want to view.

  The Transaction Details page appears.

Viewing Inbound Bank Remittance Message Details

The Message Details page lists the transactions that iPayment has sent to the back end processor in a message file and gives the message details.

To view the message details:

- Click on Message ID link to view the message details in the Message Details page.

  The Message Details page appears.

Searching for Outbound Bank Payment Transactions

To search for outbound bank payment transactions:

1. Navigate to the Outbound Payments page.
2. Enter values for the search criteria. To narrow the search criteria, you must enter as many search values as possible. You must specify a date in the Transaction Request Start Date field for the search.
3. Click Search.

   After performing a search, the Search Results table displays the summary information for transactions matching the search criteria. You can drill down to view the details for outbound bank payment transactions from the table.

> **Note:** Transactions can be sorted by batch name, document number, beneficiary name, transaction status, amount, currency, and request date. Default sorting is done by batch name first and then by request date.

- Viewing Outbound Bank Payment Transaction Details
- Viewing Outbound Bank Payment Message Details
- Viewing Outbound Bank Payment EC Batch Details

Viewing Outbound Bank Payment Transaction Details

You can view the details of a transaction from the Transaction Search Results table.

To view outbound bank payment transactions:

- Click the icon in the Details column corresponding to the transaction for which you want to view the details.

  The Transaction Details page appears with the details.

Viewing Outbound Bank Payment Message Details

The Message Details page lists the payment batches that iPayment has sent to back end payment processor in a message file and gives the message details.

To view outbound bank payment message details:

- Click on Message ID link to view the message details in the Message Details page.

  The Message Details page appears.

Viewing Outbound Bank Payment EC Batch Details

The Batch Details page lists the payment transactions that Oracle iPayment has sent to back end payment processor for a payment batch from Oracle Payables and gives the batch details.

To view outbound bank payment EC batch details:

- Click on Batch Name link to view the payment batch details in the Batch Details page.

  The Batch Details page appears.

iPayment Profile Options

For information on iPayment profile options, see iPayment Profile Options in the *Oracle iPayment Implementation Guide*.

3

Understanding Integration with Oracle Payables

This topic group provides information on the integration of Oracle iPayment and Oracle Payables.

Overview of Integration with Oracle Payables

The Oracle Payables and Oracle iPayment integration submits payment batch information from Oracle Payables to a bank server by utilizing the services of Oracle iPayment. The Oracle Payables module transfers the payment details to the Oracle iPayment repository by calling the Oracle iPayment front-end API. The Scheduler service provided by Oracle iPayment is configured to run at a predefined interval and to poll the payment data from the repository by calling the Payment Engine API. The polled data is then passed on to the bank integration servlet as an HTTP request. The servlet parses and formats the data according to the bank's format. The servlet then transfers the formatted batch files to the bank server.

Benefits of Oracle Payables - Oracle iPayment Integration

The current process for making corporate payments involves many manual steps. Oracle iPayment provides an automated process for building a payment batch, formatting and generating the output batch files, and sending the files to a bank. The automated process greatly enhances the data security, reliability, and accuracy.

The integration leverages an architecture model in which a servlet generates the formatted output file in different file formats. In order to support different payment systems, you need to build additional processor-specific servlets making the integration easily extensible.

How the Integration Works

To create payment batches and transfer the payments to banks:

1. Initiate the payment batch by entering the criteria for invoices you want to pay. The payment batch process selects invoices and then builds the payments. The process determines which invoices are paid on each payment document, and lists this information for you on the Preliminary Payment.
2. Make any necessary modifications to your payment batch, such as selecting additional invoices, deselecting invoices, and deselecting suppliers. Once modifications are complete, the modify payment batch process automatically builds the payments if any invoices were added.
3. Format payments. The format process handles the formatting of the payment documents. Oracle Payables uses the payment format program to create the layout of the checks or electronic payments.
4. Confirm the payment batch by recording the document numbers associated with each payment. During this step, Oracle Payables updates the invoice status to Paid and a document number is associated with the invoice and invoice payment.
5. Run the iPayment Scheduler for the task EFTPBATCHCLOSE or EFTPBATCHRETRY. The scheduler process generates the output file in the bank-specific format and routes it to the appropriate bank server.

Note: Oracle iPayment does not handle any reconciliation of payment between the bank and Oracle E-Business suite.

The deploying company must ensure document number uniqueness. The unique numbering reduces the manual intervention required when you do reconciliation using Oracle Cash Management.

Setup Required for Oracle iPayment Integration with Oracle Payables

The following steps describe the setup of a concurrent manager program. The system administrator does the setup during system installation.

1. Define a new concurrent program executable. Select the application name as Oracle Payables. Provide the executable file name as IBY_AP_BANKPAYMENT_PUB.IBY_AP_SUBMITBATCH.
2. Define a new concurrent program. Select the application name Oracle Payables. Select the executable created in the previous step.

   For details regarding the concurrent manager, see the *Oracle Applications System Administrator's Guide*.

The following steps describe the setup to be performed in Oracle Payables to support the integration with Oracle iPayment. For more details on each of these steps, see the *Oracle Payables User Guide*.

1. Define a new payment program in Oracle Payables. Provide the single payment program name. Select the Format Programs type. Select the concurrent program that you created in the previous step as the registered name.
2. Define a new payment format in Oracle Payables. Select the single payment format program that you created in the previous step. Define different payment formats for different currencies and payment method combinations. Link these payment formats to the same payment format program defined in the previous step.
3. Set up bank accounts. For each account, define a payment document that uses the payment format defined above.
4. In addition to the previous steps, a payment system integration must be set up. For details on the out-of-the-box solution provided with Oracle iPayment, see the *Oracle iPayment Implementation Guide*.

Payments to UK Building Societies from Oracle Payables

For payments to building societies in the UK, an additional field called Beneficiary's Roll Number needs to be passed from Oracle Payables along with the rest of the payment information. The roll number is the beneficiary's account number with the building society.

Currently Oracle ERP does not support a separate field for specifying the beneficiary roll number.

> **Note:** Payments to building societies are supported by Oracle iPayment only for UK.

Typical Usage

An organization (for example, XYZ Inc.) decides to make a payment to a building society account through its bank. XYZ Inc. wants to pay employee expenses into building society accounts.

The employee, who is the beneficiary of the payment, must provide the building society account number (beneficiary roll number) to XYZ Inc. By storing the beneficiary roll number in a supplier site level flexfield, XYZ Inc. can pass this optional field to its bank.

The existing supplier site level flexfield 'PO_VENDOR_SITES' is re-used to pass the beneficiary roll number. A new context (GB) needs to be added as per the setup below.

To define a beneficiary building society account number for individual supplier sites:

1. Choose the System Administrator responsibility.
2. Navigate to the Descriptive Flexfield Segments window.
3. Query the Supplier Site EFT Details flexfield.
4. Unfreeze flexfield definition and add a new context

   Code: GB

   Name: GB

   Description: UK Supplier EFT Information
5. Click on the Segments button to open Segments Summary window and add the following segment

   No: 1

Name: Beneficiary Roll Number

Window Prompt: Beneficiary Roll Number

Column: JGZZ_SITE_INFO4

6. Click on the Open button to open the segment and uncheck the Required check box in the Validation region. This action is necessary to indicate that this is an optional flexfield, only required for certain U.K. supplier sites.

7. Save, freeze, and then compile the flexfield definition.

The above steps are a one-time process to create a flexfield context for the beneficiary roll number.

You can now define a beneficiary roll number for your supplier sites as follows:

1. Login with the Payables responsibility.
2. Navigate to the Supplier Sites window.
3. Click on Tools->View EFT Details menu.
4. Enter Context Value as 'GB' and a value for the Beneficiary Roll Number. The context defaults to the Vat Registration Country if defined.
5. Save the flexfield.

Oracle iPayment's Single Format program picks the beneficiary roll number (if defined) for each beneficiary.

4

Transaction Reporting

This topic group provides details of the pages provided for viewing the key performance metrics such as transaction summaries, payee summaries, and other critical performance indicators.

Transaction Reporting Overview

All business intelligence information is summarized and displayed through the iPayment Transaction Reporting (TR) user interface. iPayment rolls up the key critical performance indicators across all processors, types of cards and transaction types. TR provides a tabular and graphical view of the various business trends and how they are changing. The TR user interface is browser-based and implemented using Java and Java Server Pages (JSP).

iPayment TR User Interface

This table lists the tab names and the functionality available from the iPayment TR user interface.

Note: To create a user with the Transaction Reporting (DBC) role, see ’Creating an Oracle iPayment User’ in the *Oracle iPayment Implementation Guide*.

Tab Name	Functionality
Transaction Summary	Transactions are summarized on a daily, monthly and weekly basis.
Payee Summary	Transactions are summarized on a daily, monthly and weekly basis for a selected payee

Navigating the iPayment TR User Interface

The iPayment TR user interface includes the Transaction Summary tab, the Payee Summary tab, and the Reports region space. The Transaction summary and the Payee summary tabs on the top of the page remain visible as you navigate through iPayment. The side navigation bars list the information that you can view. When you click on a navigation bar, summarized information for the selected bar appears in the report display space in the lower portion of the page.

Transaction Summary - Daily

The Transaction Summary tab opens the Credit Card Daily Business Close page.

This table describes the graphs and reports displayed in the Credit Card Daily Business Close page.

Report Name	Description
Daily Summary	Calculates and displays the number and dollar value of transactions on the current date. This is further broken down as: ■ All Transactions ■ Total Authorization Requests ■ Total Capture/Settlement Requests ■ Total Refunds/Credits ■ Total Authorizations Settled ■ Total Authorizations Outstanding ■ Total Credit Card Transactions ■ Total Purchase Card Transactions
Hourly Transaction Volume	Displays a bar graph depicting the volume of transactions for each hour on the current day.
Transaction Summary	Calculates and displays the total number of requests, total succeeded requests, total failed requests, and total pending requests on the current date for the following transaction types: ■ Authorization Requests ■ Capture/Settlement Requests ■ Refunds/Credit Requests
Transaction Failure Summary	Transactions are sorted based on the number of transactions for each ’cause of failure’. The report displays the top five causes of failure for Authorization and Settlement requests on the current date. For each cause of failure, it displays the number of failures and the dollar value of the transactions.
Card Type Summary	Summarizes the transactions by card type for the current date. The summary displays the average transaction dollar amount for credit cards, each credit card type, and purchase cards.
Processor Summary	Summarizes transactions by processor for the current date.

Report Name	Description
Transaction Risk Summary	Summarizes transactions screened for risk for the current date. Provides information such as: ■ total number of transactions screened for risk ■ percentage of transactions screened for risk ■ number of transactions identified as risky ■ percentage of transactions identified as risky

Transaction Summary - Weekly

The Weekly navigation link on the Side Panel Menu opens the Credit Card Daily Business Close - Weekly Summary page.

This table describes the reports displayed in the Credit Card Daily Business Close - Weekly Summary page.

Report Name	Description
Weekly Summary	Calculates and displays the number and dollar value of transactions during the last seven days including the current date. This is further broken down as: ■ All Transactions ■ Total Authorization Requests ■ Total Capture/Settlement Requests ■ Total Refunds/Credits ■ Total Authorizations Settled ■ Total Authorizations Outstanding ■ Total Credit Card Transactions ■ Total Purchase Card Transactions
Transaction Summary	Calculates and displays the total number of requests, total succeeded requests, total failed requests, and total pending requests for the last seven days including the current date for the following transaction types: ■ Authorization Requests ■ Capture/Settlement Requests ■ Refunds/Credit Requests
Transaction Failure Summary	Transactions are sorted based on the number of transactions for each 'cause of failure'. The report displays the top five causes of failure for Authorization and Settlement requests for the last seven days including the current date. For each cause of failure, it displays the number of failures and the dollar value of the transactions.
Card Type Summary	Summarizes transactions by card type for the last seven days including the current date. Displays the average transaction dollar amount for credit cards, each credit card type, and purchase cards.
Processor Summary	Summarizes transactions by processor for the last seven days including the current date.

Report Name	Description
Transaction Risk Summary	Summarizes transactions screened for risk for the last seven days including the current date. Provides information such as: ▪ total number of transactions screened for risk ▪ percentage of transactions screened for risk ▪ number of transactions identified as risky ▪ percent of transactions identified as risky

Transaction Summary - Monthly

The Monthly link on the Side Panel Menu opens the Credit Card Daily Business Close - Monthly Summary page.

This table describes the reports displayed in the Credit Card Daily Business Close - Monthly Summary page.

Report Name	Description
Monthly Summary	Calculates and displays the number and dollar value for transactions during the current month. This is further broken down as: ■ All Transactions ■ Total Authorization Requests ■ Total Capture/Settlement Requests ■ Total Refunds/Credits ■ Total Authorizations Settled ■ Total Authorizations Outstanding ■ Total Credit Card Transactions ■ Total Purchase Card Transactions
Transaction Summary	Calculates and displays the total number of requests, total succeeded requests, total failed requests, and total pending requests for the following transaction types for the current month: ■ Authorization Requests ■ Capture/Settlement Requests ■ Refunds/Credit Requests
Transaction Failure Summary	Transactions are sorted based on the number of transactions for each 'cause of failure'. The report displays the top five causes of failure for Authorization and Settlement requests for the current month. For each cause of failure, it displays the number of failures and the dollar value of the transactions.
Card Type Summary	Summarizes transactions by card type for the current month. The summary displays the average transaction dollar amount for credit cards, each credit card type, and purchase cards.
Processor Summary	Summarizes transactions by processor for the current month.

Report Name	Description
Transaction Risk Summary	Summarizes transactions screened for risk for the current month. Provides information such as: ■ total number of transactions screened for risk ■ percentage of transactions screened for risk ■ number of transactions identified as risky ■ percentage of transactions identified as risky

Transaction Summary - Transaction Trends

The Trends link on the Side Panel Menu opens the Credit Card Trends - Transaction Trends page.

This table describes the graphs displayed in the Credit Card Trends - Transaction Trends page.

Graph Name	Description
Total Transactions	Plots the trend of the number of transactions for the last 12 months, not including the current month.
Total Amount	Plots the trend of transaction amounts for the last 12 months, not including the current month.

Transaction Summary - Processor Trends

The Processor link on the Side Panel Menu opens the Credit Card Trends - Processor Trends page.

This table describes the graphs displayed in the Credit Card Trends - Processor Trends page.

Graph Name	Description
Total Transactions	Plots the trend of the number of transactions by processor for the last 12 months, not including the current month.
Total Amount	Plots the trend of transaction amounts by processor for the last 12 months, not including the current month.

Transaction Summary - Card Type Trends

The Card Type link on the Side Panel Menu opens the Credit Card Trends - Card Type Trends page.

This table describes the graphs displayed in the Credit Card Trends - Card Type Trends page.

Graph Name	Description
Total Transactions	Plots the trend of the number of transactions by card type for the last 12 months, not including the current month.
Total Amount	Plots the trend of transaction amounts by card type for the last 12 months, not including the current month.

Transaction Summary - Failure Trends

The Card Type link on the Side Panel Menu opens the Credit Card Trends - Failure Trends page.

This table describes the graphs displayed in the Credit Card Trends - Failure Trends page.

Graph Name	Description
Total Transactions	Plots the trend of the number of failed transactions for each failure type for the last three years, not including the current year.
Total Amount	Plots the trend of transaction amounts for each failure type for the past three years not including the current year.

Payee Summary

The Payee Summary tab opens the Select a Payee page. The Select a Payee page lists the available active payees. The user has to select a payee from this page before proceeding to view the different reports available through this tab. The selected payee becomes the default payee for the reports displayed in this tab. To change a payee, select another payee from the Select a Payee link.

Daily Summary by Payee

The Daily link on the Side Panel Menu opens the Credit Card Daily Business Close -Daily Summary by Payee page.

This table describes the graphs and reports displayed in the Credit Card Daily Business Close - Daily Summary by Payee page.

Report Name	Description
Daily Summary	Calculates and displays the number and dollar value for transactions during the current date for the selected merchant. This is further broken down as: ■ All Transactions ■ Total Authorization Requests ■ Total Capture/Settlement Requests ■ Total Refunds/Credits ■ Total Authorizations Settled ■ Total Authorizations Outstanding ■ Total Credit Card Transactions ■ Total Purchase Card Transactions
Hourly Transaction Volume	Displays the volume of transactions for each hour of the current day for the selected merchant in a bar graph.
Transaction Summary	Calculates and displays the total number of requests, total succeeded requests, total failed requests, and total pending requests for the following transaction types for the current date and for the selected merchant: ■ Authorization Requests ■ Capture/Settlement Requests ■ Refunds/Credit Requests
Transaction Failure Summary	Transactions by the selected merchant are sorted based on the number of transactions for each 'cause of failure'. The report displays the top five causes of failure for Authorization and Settlement requests for the current date. For each cause of failure, the number of failures and the dollar value of the transactions is displayed.
Card Type Summary	Summarizes transactions by card type and selected merchant for the current date. Displays the average transaction dollar amount for credit cards, each credit card type, and purchase cards.

Report Name	Description
Processor Summary	Summarizes the transactions by processor and selected merchant for the current date.
Transaction Risk Summary	Summarizes transactions screened for risk on the current date for the selected merchant. It provides information such as: ■ total number of transactions screened for risk ■ percentage of transactions screened for risk ■ number of transactions identified as risky ■ percentage of transactions identified as risky

Weekly Summary by Payee

The Weekly link on the Side Panel Menu opens the Credit Card Daily Business Close -Weekly Summary by Payee page.

This table describes the reports displayed in the Credit Card Daily Business Close - Weekly Summary by Payee page.

Report Name	Description
Weekly Summary	Calculates and displays the number and dollar value for the transactions during the last seven days including the current date for the selected merchant. This is further broken down as: ■ All Transactions ■ Total Authorization Requests ■ Total Capture/Settlement Requests ■ Total Refunds/Credits ■ Total Authorizations Settled ■ Total Authorizations Outstanding ■ Total Credit Card Transactions ■ Total Purchase Card Transactions
Transaction Summary	Calculates and displays the total number of requests, total succeeded requests, total failed requests, and total pending requests for the following transaction types during the last seven days including the current date for the selected merchant: ■ Authorization Requests ■ Capture/Settlement Requests ■ Refunds/Credit Requests
Transaction Failure Summary	Transactions by the selected merchant are sorted based on the number of transactions for each 'cause of failure'. The report displays the top five causes of failure for Authorization and Settlement requests during the last seven days including the current date. For each cause of failure, it displays the number of failures and the dollar value of the transactions.
Card Type Summary	Summarizes transactions by card type and selected merchant for the last seven days including the current date. Displays the average transaction dollar amount for credit cards, each credit card type, and purchase cards.

Report Name	Description
Processor Summary	Summarizes transactions by processor and selected merchant for the last seven days including the current date.
Transaction Risk Summary	Summarizes transactions screened for risk during the last seven days including the current date for the selected merchant. It provides information such as: ▪ total number of transactions screened for risk ▪ percentage of transactions screened for risk ▪ number of transactions identified as risky ▪ percentage of transactions identified as risky

Monthly Summary by Payee

The Monthly link on the Side Panel Menu opens the Credit Card Daily Business Close - Monthly Summary by Payee page.

This table describes the reports displayed in the Credit Card Daily Business Close - Monthly Summary by Payee page.

Report Name	Description
Monthly Summary	Calculates and displays the number and dollar value for the transactions during the current month for the selected merchant. This is further broken down as: ■ All Transactions ■ Total Authorization Requests ■ Total Capture/Settlement Requests ■ Total Refunds/Credits ■ Total Authorizations Settled ■ Total Authorizations Outstanding ■ Total Credit Card Transactions ■ Total Purchase Card Transactions
Transaction Summary	Calculates and displays the total number of requests, total succeeded requests, total failed requests, and total pending requests for the following transaction types during the current month for the selected merchant: ■ Authorization Requests ■ Capture/Settlement Requests ■ Refunds/Credit Requests
Transaction Failure Summary	Transactions by the selected merchant are sorted based on the number of transactions for each 'cause of failure'. The report displays the top five causes of failure for Authorization and Settlement requests for the current month. For each cause of failure, it displays the number of failures and the dollar value of the transactions.
Card Type Summary	Summarizes transactions by the selected merchant and card type for the current month. Displays the average transaction dollar amount for credit cards, each credit card type, and purchase cards.
Processor Summary	Summarizes transactions by processor and selected merchant for the current month.

Report Name	Description
Transaction Risk Summary	Summarizes transactions screened for risk during the current month for the selected merchant. Provides information such as: ▪ total number of transactions screened for risk ▪ percentage of transactions screened for risk ▪ number of transactions identified as risky ▪ percentage of transactions identified as risky

Transaction Summary - Transaction Trends by Payee

The Trends link on the Side Panel Menu opens the Credit Card Trends - Transaction Trends by Payee page.

This table describes the graphs displayed in the Credit Card Trends - Transaction Trends by Payee page.

Graph Name	Description
Total Transactions	Plots the trend of the number of transactions by the selected merchant for the last 12 months, not including the current month.
Total Amount	Plots the trend of transaction amounts by the selected merchant for the last 12 months, not including the current month.

Transaction Summary - Processor Trends by Payee

The Processor link on the Side Panel Menu opens the Credit Card Trends - Processor Trends by Payee page.

This table describes the graphs displayed in the Credit Card Trends - Processor Trends by Payee page.

Graph Name	Description
Total Transactions	Plots the trend of the number of transactions by the selected merchant and processor for the last 12 months, not including the current month.
Total Amount	Plots the trend of transaction amounts by the selected merchant and processor for the last 12 months, not including the current month.

Transaction Summary - Card Type Trends by Payee

The Card Type link on the Side Panel Menu opens the Credit Card Trends - Card Type Trends by Payee page.

This table describes the graphs displayed in the Credit Card Trends - Card Type Trends by Payee page.

Graph Name	Description
Total Transactions	Plots the trend of the number of transactions by the selected merchant and card type for the last 12 months, not including the current month.
Total Amount	Plots the trend of transaction amounts by the selected merchant and card type for the last 12 months, not including the current month.

Transaction Summary - Failure Trends

The Card Type link on the Side Panel Menu opens the Credit Card Trends - Failure Trends page.

This table describes the graphs displayed in the Credit Card Trends - Failure Trends page.

Graph Name	Description
Total Transactions	Plots the trend of the number of failed transactions by the selected merchant for each failure type for the last three years, not including the current year.
Total Amount	Plots the trend of the total failed transaction amounts by the selected merchant for each failure type for the last three years, not including the current year.

Index