



---

## **SDK Overview**

© 1997–2004 edocs® Inc. All rights reserved.

edocs, Inc., One Apple Hill Dr., Natick, MA 01760

The information contained in this document is the confidential and proprietary information of edocs, Inc. and is subject to change without notice.

This material is protected by U.S. and international copyright laws. edocs and eaPost are registered in the U.S. Patent and Trademark Office.

No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of edocs, Inc.

eaSuite, eaDirect, eaPay, eaCare, eaAssist, eaMarket, and eaXchange are trademarks of edocs, Inc.

All other trademark, company, and product names used herein are trademarks of their respective companies.

Printed in the USA.

---

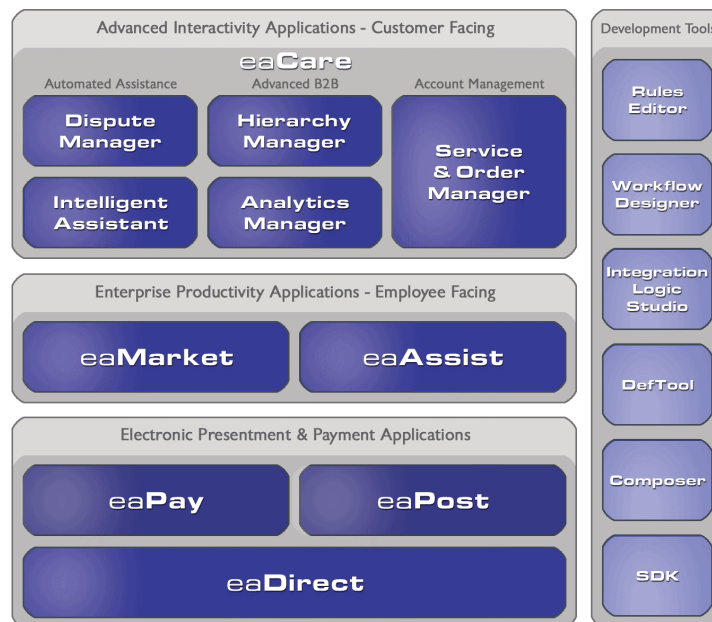
# Table of Contents

<b>Preface</b> .....	<b>5</b>
About Customer Self-Service and eaSuite™ .....	5
About This Guide .....	7
Related Documentation.....	8
Obtaining edocs Software and Documentation .....	8
If You Need Help.....	9
<b>1 Executive Summary</b> .....	<b>11</b>
What's New in 4.1 .....	11
Which Guides Do I Need?.....	11
<b>2 The eaSuite SDK Guides</b> .....	<b>13</b>
Deploying and Customizing J2EE Applications .....	13
Implementing a User Management Framework .....	14
Content Access .....	15
Auditing Data Streams .....	16
Building Custom Jobs .....	16
Line Item Dispute and Annotation .....	16
Charting.....	17
<b>3 eaSample J2EE Application</b> .....	<b>19</b>
About eaSample.....	19
Sample Code Components for the SDK .....	19
Customizing eaSample .....	20
<b>4 eaDirect SDK API Specification</b> .....	<b>21</b>
User Management.....	21
Content Access .....	22
Audit to Verify .....	22
Shell Commands for Custom Jobs.....	22
Line Item Dispute and Annotation .....	22
Hierarchy .....	23
Charting.....	23
Other .....	23



## About Customer Self-Service and eaSuite™

edocs has developed the industry's most comprehensive software and services for deploying Customer Self-Service solutions. **eaSuite™** combines electronic presentment and payment (EPP), order management, knowledge management, personalization and application integration technologies to create an integrated, natural starting point for all customer service issues. eaSuite's unique architecture leverages and preserves existing infrastructure and data, and offers unparalleled scalability for the most demanding applications. With deployments across the healthcare, financial services, energy, retail, and communications industries, and the public sector, eaSuite powers some of the world's largest and most demanding customer self-service applications. eaSuite is a standards-based, feature rich, and highly scalable platform, that delivers the lowest total cost of ownership of any self-service solution available.



eaSuite is comprised of four product families:

- Electronic Presentment and Payment (EPP) Applications

- Advanced Interactivity Applications
- Enterprise Productivity Applications
- Development Tools

**Electronic Presentment and Payment (EPP) Applications** are the foundation of edocs' Customer Self-Service solution. They provide the core integration infrastructure between organizations' backend transactional systems and end users, as well as rich e-billing, e-invoicing, and e-statement functionality. Designed to meet the rigorous demands of the most technologically advanced organizations, these applications power Customer Self-Service by managing transactional data and by enabling payments and account distribution.

**eaDirect™** is the core infrastructure of enterprise Customer Self-Service solutions for organizations large and small with special emphasis on meeting the needs of organizations with large numbers of customers, high data volumes and extensive integration with systems and business processes across the enterprise. Organizations use eaDirect with its data access layer, composition engine, and security, enrollment and logging framework to power complex Customer Self-Service applications.

**eaPay™** is the electronic payment solution that decreases payment processing costs, accelerates receivables and improves operational efficiency. eaPay is a complete payment scheduling and warehousing system with real-time and batch connections to payment gateways for Automated Clearing House (ACH) and credit card payments, and payments via various payment processing service providers.

**eaPost®** is the account content distribution system that handles all the complexities of enrollment, authentication and secure distribution of summary account information to any endpoint, while also bringing customers back the organization's Website to manage and control their self-service experience.

**Advanced Interactivity Applications** are a comprehensive set of advanced customer-facing self-service capabilities that enable the full range of business and consumer customer service activities. These sophisticated modules have the flexibility to completely customize the Customer Self-Service solution to meet vertical industry and specific company requirements.

**eaCare™** consists of a rich set of sophisticated self-service modules – Dispute Manager, Intelligent Assistant, Hierarchy Manager, Analytics Manager, and Service and Order Manager - for automated assistance, advanced business-to-business applications and account management. These capabilities come together to create a web self-service dashboard for customers to access all service offerings from a single, easy-to-use interface. eaCare's modularity accelerates time to market with components that can be deployed incrementally in a phased approach.

**Enterprise Productivity Applications** are employee-facing solutions that empower customer service representatives, sales agents, account managers, marketing managers, broker-dealers and channel partners within an organization and external partner organizations to facilitate self-service and to support assisted service. Employees leverage edocs' Customer Self-Service solution to deliver customer service, access information, create and deploy marketing and customer service content, and perform activities for the benefit of customers.

**eaAssist™** reduces interaction costs and increases customer satisfaction by enabling enterprise agents – customer service representatives (CSRs), sales agents, broker-dealers and others – to efficiently access critical account data and service-related information to effectively service customers. Through its browser interface designed especially for the enterprise agent, eaAssist enables agents to take advantage of customer-facing online capabilities to provide better service by more efficiently resolving customer account inquiries at the point of customer contact.

**eaMarket™** is the personalization, campaign and content management solution that enables organizations to increase revenue and improve customer satisfaction by weaving personalized marketing and customer service messages throughout the Customer Self-Service experience. The transactional account data that provides the foundation for a Customer Self-Service solution – such as transaction activity, service or usage charges, current task and prior service history – bring valuable insight into customers and can help optimize personalized marketing and customer service campaigns. eaMarket leverages that data to present relevant marketing and customer service messages to customers.

edocs' **Development Tools** are visual development environments for designing and configuring edocs' Customer Self-Service solutions. The Configuration Tools encompass data and rules management, workflow authoring, systems integration, and a software development kit that makes it easy to create customer and employee-facing self-service applications leveraging eaSuite.

## About This Guide

The edocs Software Developers Kit allows developers to write custom code against edocs applications. This SDK guide is intended for edocs system integrator partners, senior developers with an edocs client company, and edocs Professional Services representatives.

The SDK *SDK Overview* guide is intended for web developers supporting and extending the eaSuite with edocs APIs and customizable sample code. This overview outlines how to use each SDK guide to customize your eaSuite installation. The SDK assumes you have:

- Completed a Statement Mastering Plan
- Installed and configured eaDirect and the sample application eaSample

This SDK assumes in-depth understanding of and practical experience with:

- eaDirect system architecture, installation, deployment, application design, and administration
- Java 2 Enterprise Edition (J2EE), Enterprise JavaBeans (EJBs), servlets, and JSPs
- Packaging and deploying J2EE applications for WebLogic or WebSphere
- Directory services including the Java Naming Directory Interface (JNDI) and the Lightweight Directory Access Protocol (LDAP)
- HTML and XML, web server administration, and web browsers

## Related Documentation

A PDF version of this guide are also available.

Online	How to Access
PDF	A PDF of this guide is in the <b>EDCSbd/pdf</b> directory of your installation.

This guide is part of the eaDirect documentation set. For more information about using eaDirect, see the following guides:

Print Document	Description
eaDirect Installation and Configuration Guides	How to install and configure eaDirect in a distributed environment.
<i>Deploying and Customizing J2EE Applications</i>	How to customize J2EE web applications for deployment with the eaSuite.
<i>Data Definition Guide</i>	How to create data extraction and definition rules for an eaDirect application with the DefTool.
<i>Presentation Design Guide</i>	How to design data presentment for an eaDirect application with the Composer.
<i>Data Presentation Production Guide</i>	How to set up and run a live eaDirect application in a J2EE environment.
Software Developer Guides (SDK) and eaDirect API Specification	How to design and implement your own custom J2EE web application for eaDirect.

## Obtaining edocs Software and Documentation

You can download edocs software and documentation directly from Customer Central at <https://support.edocs.com>. After you log in, click on the Downloads button on the left. When the next page appears, you will see a table displaying all of the available downloads. To search for specific items, select the Version and/or Category and click the Search Downloads button. If you download software, an email from edocs Technical Support will automatically be sent to you (the registered owner) with your license key information.

If you received an edocs product installation CD, load it on your system and navigate from its root directory to the folder where the software installer resides for your operating system. You can run the installer from that location, or you can copy it to your file system and run it from there. The product documentation included with your CD is in the Documentation folder located in the root directory. The license key information for the products on the CD is included with the package materials shipped with the CD.



## If You Need Help

Technical support is available to customers who have valid maintenance and support contracts with edocs. Technical support engineers can help you install, configure, and maintain your edocs application.

edocs provides global Technical Support services from the following Support Centers:

**US Support Center**

Natick, MA  
Mon-Fri 8:30am – 8:00pm US EST  
Telephone: 508-652-8400

**Europe Support Center**

London, United Kingdom  
Mon-Fri 9:00am – 5:00 GMT  
Telephone: +44 20 8956 2673

**Asia Pac Rim Support Center**

Melbourne, Australia  
Mon-Fri 9:00am – 5:00pm AU  
Telephone: +61 3 9909 7301

**Customer Central**

<https://support.edocs.com>

**Email Support**

<mailto:support@edocs.com>

When you report a problem, please be prepared to provide us the following information:

- What is your name and role in your organization?
- What is your company's name?
- What is your phone number and best times to call you?
- What is your e-mail address?
- In which edocs product did a problem occur?
- What is your Operating System version?
- What were you doing when the problem occurred?
- How did the system respond to the error?
- If the system generated a screen message, please send us that screen message.
- If the system wrote information to a log file, please send us that log file.

If the system crashed or hung, please tell us.





---

## Executive Summary

### What's New in 4.x

- [Deploying and Customizing J2EE Applications](#): **now ships with eaDirect**
- [Implementing a User Management Framework](#): Javadoc updates only
- [Content Access](#): Javadoc updates only
- [Auditing Data Streams](#): Javadoc updates only
- [Building Custom Jobs](#): Javadoc updates only
- [Line Item Dispute and Annotation](#): **now supported in eaCare**; Javadoc updates only
- [Charting](#): Javadoc updates only
- [eaDirect SDK API Specification](#): **new Hierarchy APIs, Javadoc updates**

### Which Guides Do I Need?

#### Summary: Deploying and Customizing J2EE Applications

This guide now ships with **eaDirect**. It is noted here as required for the SDK.

- The components of a J2EE web application for eaDirect, which can include customized JSPs, HTML pages, EJBs, and Java APIs and class files
- The components of an eaDirect application (dataset), which includes a data source file, data definition and application logic files, and HTML templates
- Introduction to the customizable web application **eaSample**
- How to build a custom web application for eaDirect by customizing sample components of **eaSample** and **eaTraining**
- How to deploy your custom web application to your application server

#### Summary: Implementing a User Management Framework

- Concepts in naming services and directory access

- Core decisions surrounding customer login and data access
- How to plan and select a directory framework for user management
- How to build and manage a hierarchical user profile schema with the edocs Common Directory Access (CDA) framework
- How to implement non-directory access enrollment models

## Summary: Content Access

- How to plan and design data access for retrieving and presenting statements
- How to customize summary and detail web views
- How to record and present web-time activity, such as when a customer last viewed a statement

## Summary: Auditing Data Streams

- How to create an audit trail to review, accept, or reject a volume of statement data before presentment
- How to query which applications are deployed on an eaDirect server

## Summary: Building Custom Jobs

- How to add custom job types to the edocs Command Center with shell commands
- How to customize and schedule Command Center jobs and tasks

## Summary: Line Item Dispute and Annotation

This guide now supports features of **eaCare**. It is noted here as required for the SDK.

- How to extract and retrieve line item detail
- How to dispute all or part of a line item
- How to add annotations to line items

## Summary: Charting

- How to present statement data as a graphical chart in a dynamic HTML page
- How to customize the eaDirect charting servlet

# 2

---

## The eaSuite SDK Guides

### Deploying and Customizing J2EE Applications

This guide now ships with eaDirect. It is noted here as required for the SDK.

#### **1 About edocs Sample Applications**

Describes the features and components of the J2EE, web, and eaDirect applications included with your eaDirect installation.

#### **2 J2EE Applications for eaDirect**

Describes the features and components of eaSample and eaTraining, the two customizable sample web applications included with eaDirect.

#### **3 Renaming eaSample to a New J2EE Application**

Describes how to edit deployment descriptors for the EAR, WAR, and EJB components and how to repackage and deploy the EAR File

#### **4 Creating Your Own Custom Web Application for eaDirect**

Discusses typical customizations of sample J2EE applications, including how to define a custom enrollment model, define custom servlets, and customize the WAR File. Implement JSP validation, and deploy in directory mode for WebLogic.

Two Appendices list components of EAR and WAR samples and eaDirect Application Datasets.

# Implementing a User Management Framework

## 1 About User Management

Defines the concepts and goals of the edocs User Management Framework and identifies the core decisions surrounding customer login and data access needed to plan and design your user management framework. Describes components of the Sample J2EE Applications shipped with eaDirect, particularly Java Server Pages (JSP) and servlets in the sample web applications. Describes how to authenticate user identity and add and modify user accounts using the **IAccount** interface for edocs enrollment. Also defines the core interfaces and methods in the package **com.edocs.enrollment.user**, including **IAccount** and **IAccountResolver**, and describes methods for contexts, objects and attributes, searching, and authentication.

## 2 Introduction To Directory Access

provides basic concepts and definitions for , directory access services, naming systems, naming services, and namespaces, directory contexts, directory information trees (DIT), directory entries, objects, and attributes, directory schema, and distinguished names. Choosing a Directory Access Interface compares features of the Java Naming and Directory Interface™ (JNDI), Lightweight Directory Access Protocol (LDAP), and Common Directory Access (CDA) Interface.

## 3 Using the Common Directory Access (CDA) Framework

Introduces the edocs user management framework shipped with eaDirect, including the edocs default CDA schema and DIT used in the eaTraining web application. Discusses how to deploy and configure eaTraining as a template with National Wireless. Defines how eaTraining uses CDA and outlines the eaTraining CDA process flow. Describes how to modify the CDA EJB for your application and how to use **IAccount** with CDA to obtain attributes and their values, manage user levels, and search for attributes.

## 4 Using the CDA Client

Using the CDA Client introduces the edocs tool for creating and managing directory schemas. Describes how to start the CDA Client and parse CDA commands, navigate within a DIT, and work with directory contexts and user attributes. Also discusses how to integrate with existing user management systems by exporting a schema as LDIF and importing an LDIF schema into CDA.

## 5 Using a Non-Directory Access Implementation

Discusses implementing the edocs user management framework outside of CDA. Describes how to deploy and configure **UMFsample** as a Template with National Wireless. Defines how **UMFsample** does enrollment and outlines the **UMFsample** Enrollment Process Flow. Also discusses how to create a custom an application based on **UMFsample** by modifying the enrollment source files, defining a custom enrollment EJB, and building the new custom version of **UMFsample**

Two Appendices suggest additional background reading and describe the User Management Framework API packages shipped with eaDirect.

## Content Access

### 1 Planning Your Content Access Interface

Planning Your Content Access Interface defines the concepts and goals of content access and provides basic concepts about XML, XSL, and XSLT as used with eaDirect.

### 2 Introduction to edocs Content Access

Describes the new XML views and jobs for eaDirect and the XML Templates available to customize input and output.

### 3 XML and eaDirect

Discusses how eaDirect uses XML, how to map a DDF to XML, and provides further reading about XML, XSL, and XSLT.

### 4 Extracting Detail Data to the Database

Describes the Detail Extractor Job and how to customize the XML templates provided.

### 5 Transforming Data with XSLT

Describes the XSLT View Type and how to customize the XML templates provided.

### 6 Extracting Data with XML Queries

Describes the XML Query View and how to customize the XML templates provided.

### 7 Application Programming Interfaces (API) for Content Access

Provides a description of package `com.edocs.app.user` methods, including the `User` and `UserMain` methods.

### 8 Using Content Access APIs

Describes how to use Content Access APIs by calling user methods in correct sequence to retrieve and present statement summaries, retrieve and present statement detail, retrieve and present XML, and record and present web-time activity.

## **9 Element ID and Composition Hints**

Defines Element ID and Composition Hints, the rich language of XML metadata provided with eaDirect. Defines syntax and tag attributes for Element ID and defines values and DTD for the Composition Hints language.

Appendix A lists sample code for the National Wireless XML Templates for Detail Extractor, XSLT View, XML Query View, and an example DDF to XML Mapping.

# **Auditing Data Streams**

## **1 Introduction to Auditing Data Streams**

Describes when and why to audit data streams of online statements before presentment with the Verify API.

## **2 Using the Verify API**

Defines Verify methods and signatures to retrieve a list of all applications, retrieve a list of indexed volumes, retrieve a list of account numbers, retrieve account summary information, accept or reject an indexed volume, and update summary information.

# **Building Custom Jobs**

Describes when and why to customize a Command Center job type with the Shell Command Task, and provides examples of how to script and configure your new job type.

# **Line Item Dispute and Annotation**

This guide now supports features of eaCare. It is noted here as required for the SDK.

## **1 Introduction to Line Item Dispute and Annotation**

Defines goals of adding disputes and annotations to eaDirect statement data, and compares the features of disputes with annotations.

## **2 Components of Line Item Dispute and Annotation**

Provides an architectural overview of dispute and annotation features and components and outlines the task flow of configuration, composition, production, and web application tasks. Also gives step-by-step procedures for viewing examples of National Wireless sample data in the eaSample web application, both shipped with eaDirect.



### **3 Configuring Dispute and Annotation Services**

Defines JMS settings and database tables required for dispute and annotation.

### **4 Composition and Production for Dispute and Annotation**

Provides an architectural overview of composition and production components, including Element ID, edit XSLT templates for Detail Extractor and XML Query views, how to create and configure a Detail Extractor job and publish three required XML Query dynamic web views.

### **5 Web Application Components for Dispute and Annotation**

Provides an architectural overview of web application components, including the Manage Statement JSPs for line item detail, dispute, and annotation. Describes how to use Apache XTags with Dispute and Annotation JSPs.

### **6 Application Programming Interfaces (API) for Dispute and Annotation**

Gives an overview of data flow for Annotation and Dispute Services. Defines the main classes and methods in the packages `com.edocs.direct.dispute` and `com.edocs.direct.annotation`, including the `submit` and `getDocument` methods.

### **7 Using the Dispute and Annotation APIs**

Implementing Line Item Dispute and Annotation discusses the APIs needed to submit, retrieve, update, and delete disputes and annotations, including an overview of retrieving line item detail with the Content Access API `com.edocs.user`.

Appendix A lists eaSample JSPs and National Wireless XML templates for dispute and annotation.

## **Charting**

### **1 Introduction to Charting**

Gives an overview of data flow and components for formatting statement data as a graphical chart in a dynamic HTML page.

### **2 Configuring Charting for Your Server**

Discusses the procedures for configuring display devices and permissions, including how to configure a “headless” server (without a dedicated display) for charting.

### **3 Composing Charts in Statements**

Describes how to use the Composer to create an Application Logic File (ALF) and a chart properties file to display charts in statements, and how to simulate the appearance of your published chart at runtime with the Simulator tool.

### **4 Customizing Chart Properties**

Describes how to adjust chart settings in ALF attributes and chart properties, and how to preview your customized chart.

### **5 Publishing Charts**

Describes how to create and configure a Chart view in the Command Center, and how to test chart viewing in your web application.

### **6 Designing Custom Charts with the Charting Servlet**

Discusses the default charting servlet, `charting.java`, shipped with eaDirect and provides suggestions and tips for writing your own custom chart servlet and integrating it into your web application.

### **7 Troubleshooting Charts**

Lists solutions to common problems with charting.

### **8 Application Programming Interfaces (APIs) for Charting**

Defines the main classes and methods in the package `com.edocs.app.charting`, including the `chartClient` and `chartData`, `Charter`, `Publisher`, and `Simulator` classes.

# 3

---

## eaSample J2EE Application

### About eaSample

The eaSample EAR file contains sample code components to demonstrate the features of eaDirect. For a list of the components in eaSample, please see *Deploying and Customizing J2EE Applications*.

### Sample Code Components for the SDK

#### Deployment

`validation.jsp` provides a code fragment to be included in other JSPs for data validation.

#### User Management and Content Access

The eaSample JSP pages allow a user to enroll, log in, view a summary of online statements, and view details of a selected statement.

#### Line Item Dispute and Annotation

eaSample also includes sample code components for implementing line item dispute and annotation. For details of these components and discussion of how to customize these features, see the SDK guide: *Line Item Dispute and Annotation*.

#### Auditing Data Streams

For details of how to customize JSPs to audit data volumes before presentment with the Verify API, see the SDK guide: *Auditing Data Streams*.

#### Custom Jobs and Tasks

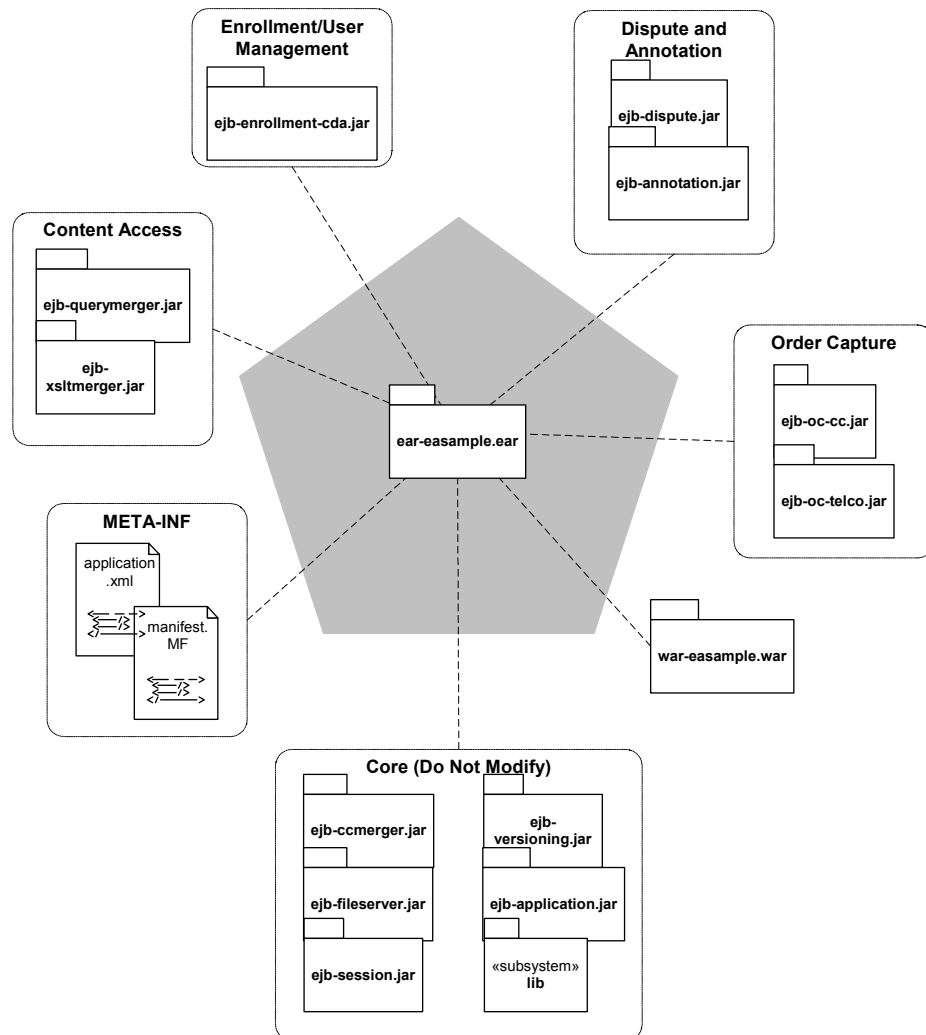
For details of how to use the Shell Command API to customize jobs and tasks, see the SDK guide: *Custom Jobs and Tasks*.

## Online Payment with eaPay

eaSample does not integrate online payment features, and links to payment options are placeholder stubs only. For details of payment components and discussion of how to customize online payment, see the SDK guide: *Customizing eaPay*.

## Customizing eaSample

Depending on what you plan to customize, you will work with one or more of the JAR files shown in this diagram.



# 4

---

## eaDirect SDK API Specification

### User Management

`com.edocs.app` provides the interface `LoginRequired` and base servlet classes for eaDirect. Class `App` is the base class for all eaDirect application servlets. Class `AppConstant` holds string fields for request attributes. Class `AppServlet` is a servlet that extracts and dispatches the name of the APP requested. Class `LoginApp` is the base class for login application servlets, providing session-based management of `IAccount`.

`com.edocs.app.enrollment` provides utility classes for enrollment and default implementations of Login and Logout servlets.

`com.edocs.enrollment` provides a set of utility classes and exceptions to support user authentication with `IAccount`. Interface `EnrollmentConstants` supports implementation of the class `AccountUtils` in `com.edocs.enrollment.user`. Class `Encrypt` provides a method to encrypt a user ID with a password. Class `NameValue` represents an attribute as a name/value pair, manipulated by the helper class `Parameters`.

`com.edocs.enrollment.user` provides interfaces and classes for edocs user management framework. Includes two primary interfaces. `IAccount` defines directory services for user enrollment and authentication. `IAccountResolver` provides a batch interface for retrieving attributes for a directory context, for example to return the e-mail address for an account number.

`com.edocs.enrollment.user.jndi` provides an implementation of the `IAccount` interface, which accesses edocs' Common Directory Access (CDA). Includes the interface `JNDIAccountAttributes` and classes `JNDIAccount`, `Login`, and `JNDIAccountResolver`. Intended for advanced application development.

`com.edocs.jndi.cda` provides an interface and classes for implementations of Common Directory Access (CDA), the edocs limited implementation of an LDAP-like JNDI service provider. Includes the interface `CDAConstants` and classes `CDANameParser`, `CDAFactory`, and `cdaURLContextFactory`.

`com.edocs.jndi.cda.cli` provides the command-line interface to CDA, edocs' Common Directory Access. This tool provides the ability to create and manage directory schemas.

`com.edocs.services.session` provides the interface `ISession` and classes to obtain and return the edocs Session object of a servlet when `com.edocs.app` calls `App.getSession`.

## Content Access

`com.edocs.app.user` provides classes `User` and `UserMain` for access to statement summary and detail. The `User` class is the content access interface to the eaDirect core. Its methods retrieve and send statement data for a given user account, as well as sorting, subtotaling, and updating optional fields at presentment. Class `UserMain` implements two interfaces. The interface `com.edocs.app.LoginRequired` informs the edocs servlet framework that the requesting client must authenticate itself before accessing `UserMain`. The interface `Servlet` defines basic methods that any servlet must implement. Its methods `doGet` and `doPost` support, respectively, HTTP requests for GET and POST.

## Audit to Verify

`com.edocs.app.verify` provides the `Verify` class and methods for auditing indexed volumes of data before releasing them for presentment. `getIndexedList` retrieves a list of indexed volumes available for audit, while `getAccountList` retrieves all the account numbers in a volume. `getDDNList` retrieves all DDNs. Two signatures of `getHitList` retrieve all Description items either for a given volume, or for a given account. `acceptIndexedVolume` or `rejectIndexedVolume` respectively accept or reject a volume for presentment to customers. `updateDescriptionInfo` supports updates to the optional information field (`Y_#`) on a statement page.

## Shell Commands for Custom Jobs

`com.edocs.tasks.shellcmd` provides the `ShellCmdTask` class as a task that executes an external shell command, for example to create custom Command Center jobs.

## Line Item Dispute and Annotation

This API now supports features of `eaCare`. It is included in the *eaDirect SDK API Specification*.

`com.edocs.direct.annotation` provides the `Annotation` class and methods `submit` and `getDocument` to allow users to create, update, or cancel an annotation to line item detail, and to retrieve annotation data from the database.

`com.edocs.direct.dispute` provides the `Dispute` class and methods `submit` and `getDocument` to allow users to submit a dispute to line item detail, and to retrieve disputes from the database.

## Hierarchy

This API now supports features of **eaCare**. It is included in the *eaDirect SDK API Specification*.

`com.edocs.hierarchy` provides an interface to define the values of constants for hierarchy fields, and utility classes for creating a hierarchy as a directory information tree (DIT).

`com.edocs.hierarchy.app` provides servlet classes to extend and override the example servlet classes in `com.edocs.app` for user login and content access to account data mapped to the hierarchy.

`com.edocs.hierarchy.navigate` provides an abstract interface and sample implementation class to filter and display content in the Hierarchy Manager.

`com.edocs.hierarchy.render` provides an interface and classes to design and display the user interface of the hierarchy console as HTML.

`com.edocs.hierarchy.taglib` provides a custom tag library of JSP tags for presenting hierarchy data.

## Charting

`com.edocs.app.chart` Charting in eaDirect is achieved using third party **Kavachart**'s charting utility, an edocs wrapper to set and interpret chart properties, and by extending the eaDirect publisher to accept chart views.

## Other

`com.edocs.common.web.validation` Implements the class `ValidatorBean` to support validation of input to JSP pages. Bean methods capture, set, validate, and write the list of legal and illegal parameter names and values in a `ServletRequest` object.