



## What's New in eaDirect 3.4

eaDirect is a member of the eaSuite™ product line

V3.4  
Document ID: DIGN-07-3.4-01  
Date Published: 12.16.02

© 1997–2002 edocs® Inc. All rights reserved.

edocs, Inc., Two Apple Hill, 598 Worcester Road, 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, eaProcess, eaService, 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>7</b>
About Customer Self-Service and eaSuite™ .....	7
Related Documentation .....	10
If You Need Help .....	11
<b>1 New Features in eaDirect</b> .....	<b>13</b>
New in eaDirect 3.4 .....	13
Reporting and Analytics with eaCare.....	13
Running Jobs Concurrently.....	13
Defining Extended Functions For Group Fields.....	14
Positioning Starting Columns For DB Extract Files .....	14
DDN to Datasource Mapping .....	14
New in eaDirect 3.3 .....	14
Native XML Input.....	14
New in eaDirect 3.2 .....	15
Sub Account Indexing .....	15
Enterprise Profile Management with the Hierarchy Console .....	15
Cut, Copy, and Paste in DefTool .....	16
New Custom Post-Conversion Builder.....	16
<b>2 Reporting and Analytics</b> .....	<b>19</b>
About Reporting and Analytics.....	19
Creating and Configuring a Report Job.....	19
Installing and Configuring Reporting.....	20
Creating an Indexer Job For Reporting.....	20
Creating and Configuring a Report Job .....	26
Indexing the Generated Report Data .....	30
Testing Reporting.....	30
<b>3 Running Jobs Concurrently</b> .....	<b>33</b>
Overview of Concurrency.....	33

Summary Job Information on the Main Console .....	34
Managing Individual Job Instances .....	35
Number of Rows On the Task Status Page .....	37
Canceling and Retrying Failed Jobs .....	39
Configuring and Scheduling Concurrent Jobs .....	41
Multiple Statement Processing .....	43
<b>4 Other New Features in eaDirect 3.4 .....</b>	<b>45</b>
Defining Extended Functions For Group Fields .....	45
Positioning Starting Columns For DB Extract Files .....	45
DDN to Datasource Mapping .....	45
<b>5 Native XML Input .....</b>	<b>47</b>
About Native XML Input .....	47
Overview of XML Input Workflow .....	48
About XML Input Files for eaDirect .....	49
Enabling Your Web Application for XML Input .....	49
Creating and Configuring an XML Loader job .....	50
Task 1: Scanner .....	55
Task 2: XMLLoader .....	56
Task 3: XIR Loader .....	56
Task 4: AutoIndexVolAccept .....	57
Creating and Configuring an XML Email Notification Job .....	57
Task 1: IVNScanner .....	61
Task 2: XMLMailNotification .....	62
<b>6 Sub-Account Indexing .....</b>	<b>63</b>
About Sub-Account Indexing .....	63
Sample Application For Sub-Account Indexing .....	64
Setting Up Sub-Account Indexing .....	64
Creating a DDF for Indexing and Sub-Account Presentment .....	67
Modifying a Group Field Or Subkey Properties .....	73
Deleting a Group Field Or Subkey .....	73
Creating an ALF for the Sub-Account View in Composer .....	74
Applying Sorting, Filtering, and Other Features Within Sub-Accounts .....	77
Purging Sub-Account Index Data .....	78
APIs for Sub-Account Indexing .....	78
com.edocs.app.user .....	78
com.edocs.app.verify .....	79

<b>7</b>	<b>Enterprise Profile Management.....</b>	<b>81</b>
	About the Hierarchy Console .....	81
	Setting Up the Hierarchy Console.....	83
	Adding Attributes to the CDA Schema.....	84
	Deploying the Hierarchy Web Application .....	86
	Indexing Hierarchy Sample Data .....	86
	Working with Hierarchies .....	87
	SuperAdmin and Admin Roles and Privileges .....	87
	Logging In to the Hierarchy Console.....	87
	Navigating the Hierarchy Console .....	91
	Creating Hierarchy Folders .....	93
	Editing Hierarchy Folders.....	94
	Deleting Hierarchy Folders .....	95
	Enrolling Users in the Hierarchy .....	95
	Assigning Accounts Into Hierarchy Folders .....	97
	Assigning Users Into Hierarchy Folders.....	99
	Searching for Accounts in the Hierarchy.....	100
	Viewing Accounts For a User in the Hierarchy .....	100
	Customizing the Hierarchy Console .....	100
	Application Programming Interfaces (APIs).....	101
	com.edocs.hierarchy .....	101
	com.edocs.hierarchy.app .....	101
	com.edocs.hierarchy.render .....	101
	com.edocs.hierarchy.taglib .....	101
<b>8</b>	<b>Cutting, Copying, and Pasting DDF Entities .....</b>	<b>103</b>
	About Cutting, Copying, and Pasting in a DDF .....	103
	Page Styles .....	104
	Fields, Group Fields.....	104
	Markers .....	104
	Tables, Columns .....	105
	Groups .....	105
	How to Move or Copy DDF Entities .....	105
<b>9</b>	<b>Custom Post-Conversions.....</b>	<b>109</b>
	About Custom Post-Conversions .....	109
	Date and Time Formats .....	110

Reusing Custom Post-Conversions .....	110
Creating Custom Post-Conversions .....	110
Editing a Post-Conversion .....	115
Removing a Custom Post-Conversion From a Field Or Column .....	115
Sharing Custom Post-Conversions With Other Applications .....	116
Date and Time Format Constraints .....	119
Examples of Custom Post-Conversions .....	119
Explanation of First Example .....	121
Explanation of Second Example .....	122
<b>Index .....</b>	<b>123</b>

## Preface

### About Customer Self-Service and eaSuite™

#### eaSuite

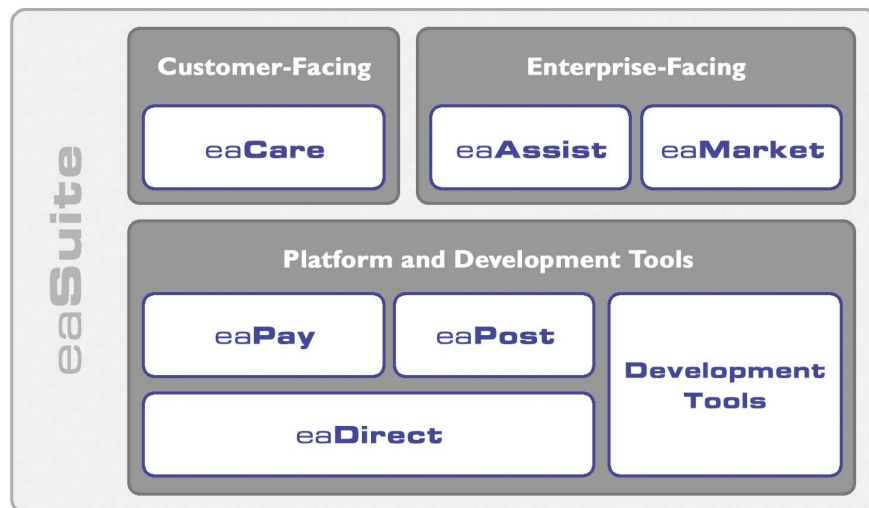
edocs has developed the industry's most comprehensive software and services for deploying Customer Self-Service solutions. eaSuite™ combines electronic presentment and payment (EPP), order management, knowledge management, personalization and application integration technologies to create an integrated, 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 designed to support how organizations approach designing and deploying Customer Self-Service applications:

**Customer-Facing Solutions** present customers with the sophisticated functionality to meet customers' self-service needs. eaSuite offers a full set of capabilities to enable the range of business and consumer customer service activities, along with the flexibility to completely customize the solution to meet vertical industry and specific company requirements.

**Enterprise-Facing Solutions** empower employees within an organization and external partners to leverage the edocs platform to facilitate self-service and to support assisted service. Customer service representatives (CSRs), sales agents, account managers, marketing managers, broker-dealers and channel partners all play a role in delivering customer service, creating content, accessing information and performing activities for the benefit of customers.

**Platform and Development Tools** are designed to meet the rigorous infrastructure demands of the most technologically advanced organizations. These components of the eaSuite power edocs solutions with the functionality and development tools necessary to make account data available, and to create the customer- and enterprise-facing applications that enable customer self-service.





**eaAssist**

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**

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.

**eaDirect**

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**

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**

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.

### **Development Tools**

eaSuite Development Tools™ are visual development applications that provide intuitive graphical user interface (GUI) environments for designing and developing Customer Self-Service solutions. The Development Tools encompass data management, workflow authoring, rules management and accounts receivable integration, as well as a full Software Developers Kit for custom application development.

## **Related Documentation**

A PDF version of this guide are also available.

<b>Online</b>	<b>How to Access</b>
A PDF of this guide	A PDF of this guide is available on the eaDirect CD-ROM.

This guide is part of the eaDirect documentation set. For more information about implementing your eaDirect application, see one of the following guides:

<b>Print Document</b>	<b>Description</b>
eaDirect Installation and Configuration Guide	How to install eaDirect and configure it in a distributed environment.
eaDirect User's Guide	How to create data extraction and design files for an eaDirect application using the DefTool and Composer composition tools.
eaDirect Production Guide	How to set up and run a live eaDirect application in a J2EE environment.
eaDirect Developer's Guide	Provided with the eaDirect Software Developer's Kit (SDK), describes eaDirect application server components and related applications; defines the public interfaces for customizing and extending the functionality of an eaDirect application.

Add-on eaSuite products eaPay, eaPost, eaMarket, and eaXchange provide their own documentation.

## 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.

To reach the U.S. Service Center, located in Natick, MA (Monday through Friday 8:00am to 8:00pm EST):

- Telephone: 508.652.8400
- Toll Free: 877.336.3362
- E-support: [support.edocs.com](http://support.edocs.com) (This requires a one-time online registration)
- E-mail: [support@edocs.com](mailto:support@edocs.com)

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

## Preface

- 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.

# New Features in eaDirect



This document supplements the currently published guides for the eaSuite. For more information, please see the *eaDirect User's Guide*, *Production Guide*, and *Software Developer's Kit (SDK)*.

## New in eaDirect 3.4

### Reporting and Analytics with eaCare

- Create phone call usage reports from indexed data with the new Report job.

### Running Jobs Concurrently

- Schedule multiple thread-safe jobs to run concurrently within an application, leveraging computing power to complete jobs faster.
- Specify the maximum number of concurrent instances to allow for each job: 5, 10, 15, or 20.
- Start up to 20 instances of a job in the same scheduled run, then monitor and manage instances on the Task Status page.
- Process multiple statements at one time with the new **Report** job. The **StatementScanner** task runs a job instance for each statement in the input file, optimizing processing time for a file with many documents.

## Defining Extended Functions For Group Fields

- Use Composer to create field elements that sort, filter, page, combine, calculate arithmetic, and analyze business conditions on group fields.

## Positioning Starting Columns For DB Extract Files

- When using DefTool to create a new DDF for a DB Extract input file, you can now specify a starting column position if the page delimiter is a string.

## DDN to Datasource Mapping

- Specify a custom datasource EJB for each eaDirect application directly in the Command Center, without hand-editing deployment descriptors and redeploying your web application.

# New in eaDirect 3.3

## Native XML Input

Since eaDirect 3.0, you have been able to extract and publish your data as XML using schemas, stylesheets, and other XML design files. Native XML input completes the end-to-end workflow by accepting input data in XML format and processing it with XSD schemas, attributes, and XSL stylesheets.

### Benefits

- This scalable, persistent edocs solution lets you manage large volumes of XML data in real time, while saving system costs of processing and composing raw XML.

## New in eaDirect 3.2

eaDirect 3.2 enhances its B2B capabilities with the addition of sub-account indexing and the Hierarchy Console for improved access and reporting over organizational account structures. For the business user, these enhanced capabilities make account data easier to analyze and act on, consequently making account information more meaningful and more valuable. eaDirect 3.2 also lets you cut, copy, and paste DDF elements in DefTool for additional efficiency, and adds tools for custom post-conversion of statement elements.

### Sub Account Indexing

- Increased flexible optimization controls specifically for massive B2B data volumes
- Sub-account indexing speeds access through large data volumes directly to specific account sub-structures
- The Development Environment and Command Center updated for new configuration support

#### Benefits

- Increased control for configuring data sources in multiple application environments; configure applications for B2B or B2C data sources or mix configurations for consolidated applications
- Delivers additional reporting options for cross-invoice reporting, summary reports, and trending at the summary level

### Enterprise Profile Management with the Hierarchy Console

- New Web console interface for creating and managing organizations
- Manage structures of accounts and users via a Web Console, create groupings of sites, departments or regions

- Pinpoint users and accounts with search and return lists
- Includes user navigational access control in one console for easy management
- Supports upload and export of data for updates and reuse

### **Benefits**

- Allows operations staff, customer service representatives or customer organizations themselves to create and manage their organizational hierarchy
- Visual and intuitive Web console supports a broad spectrum of skills, using a flexible and open format that can be customized
- Provides one Web console for a view of the organizational structure of accounts and users
- Supports custom mappings and attributes with full create, update and maintenance of mappings, useful for extensions like Device IDs and tracking

## **Cut, Copy, and Paste in DefTool**

You can now cut, copy, and paste Page styles, Fields (including group fields), Markers, Tables and columns, and Groups within a DDF or to another DDF:

### **Benefits**

Using cut, copy, and paste saves time and reduces errors when building multiple application views.

## **New Custom Post-Conversion Builder**

- Development Environment updated with new formatting specification builder
- Default list of conversions still provided, but now it's easy to add your own



- Create your new conversion routines with the "conversion builder"
- Add, save and re-use conversion routines

### **Benefits**

- Frees application developers to create their own format conversions
- Especially useful for internationalization of applications where presentation requirements vary widely.



## About Reporting and Analytics

The eaSuite includes powerful new tools for customer self-care with eaCare. The new Report job in the Command Center allows you to analyze customer data and present usage reports in customer statements.

## Creating and Configuring a Report Job

The Report job creates phone call usage reports from indexed data for the following criteria:

- Top 20 most expensive calls
- Top 20 longest calls
- Top 20 most expensive called numbers
- Top 20 most called numbers
- Top 20 most called countries
- Cost sum by call types
- Cost sum by months
- Cost sum by time periods

## Installing and Configuring Reporting

Perform the following steps to install and configure your system to test reporting:

1. Create a new application to index the sample data. Sample DDFs, ALFs, and templates are provided using NationalWireless data under the *EDCSbd/samples/NW\_Reports* directory.
2. Create an Indexer job, and publish a DDF that contains the call detail table on which your report will be based on. Some of those details will be selected by the report description XML file (RDX) for the report data.
3. Publish an XML dynamic web view using the same DDF, because the Report job translates the data into XML format as part of creating the reports.
4. Edit the sample report description XML file (RDX), to select the columns you want in your report.
5. Run the Report job, which creates a subdirectory for each of the eight report types and puts data into them that will be used to create reports.
6. Create new applications for each reporting type. Then, under each application, run the Indexer job to index the data for each report type, in the appropriate directory. This allows eaSample to display the reports to the user(s).
7. Test your reports by viewing them in eaSample.

## Creating an Indexer Job For Reporting

### To create an Indexer job for Reporting:

1. The Create New Job screen appears automatically after you create a new application. Otherwise, click the application name on the Main Console, then click the **Add New Job** button. eaDirect displays the Create New Job screen:

## Create New Job: myReport

When creating a job for an application that has just been created, you will need to publish the ALF, DDF and associated HTML template files. For adding additional jobs to an existing application, publishing the files is often not necessary, unless you wish to modify or create new versions of these files.

**STEP 1** Name new job and select a job type.

<b>Job Name:</b>	<input type="text" value="Indexer"/>
<b>Job Type:</b>	<input type="text" value="Indexer"/>

**STEP 2** Publish application/job files and templates.

**Launch Publisher**

**STEP 3** Configure and Schedule job.

**Configure Job and Continue** **Help**

© Copyright 1997-2002 edocs®, Inc. All Rights Reserved.  
edocs is Reg. U.S. Pat. & Tm. Off. [Privacy Policy](#)

2. Enter a meaningful name for the Indexer job. The job name can contain alphanumeric characters and underscores, but no spaces. The Indexer job name cannot start with a numeric.
3. Select the Indexer job type from the drop-down menu.
4. Click **Launch Publisher**.

**edocs**

**Select a Version Set Type**

**Dynamic Web Views**

Job Type	Number of Auxiliary Files
HTML	0   1   2   3   <a href="#">more</a>
CSV	0
XML	0
CHART	0
XSLT	0
XMLQuery	0

**Batch Jobs**

Job Type	Number of Auxiliary Files
Detail Extractor	0
Email Notification	0   1   2   3   <a href="#">more</a>
HTML Output	0   1   2   3   <a href="#">more</a>
Indexer	0
XML Output	0

www.edocs.com

- Under Batch Jobs, next to Indexer, click 0 (Number of Auxiliary files). Publisher displays the Create a Version Set For Indexer screen:

**Create a version set for Indexer**

Application:

View Name: Indexer

DDF File:

© Copyright 1997-2002 edocs®, Inc. All Rights Reserved.  
edocs is Reg. U.S. Pat. & Tm. Off. [Privacy Policy](#)

6. Select the application name from the drop-down list. For example, myReport.
7. Browse and select the DDF file for the Indexer job. The sample DDF file is *EDCSbd/samples/NW\_Reports/NW\_Reports.ddf*.
8. Click **submit**. Publisher displays the Submission screen with details about the DDF file:

**Submission**

**Application:** esReport  
**View Type:** DOC\_CONFIG  
**View Name:** Indexer  
**Timestamp:** Fri Dec 13 13:07:40 GMT-05:00 2002

**This version set contains the following files:**

[esReport/DOC\\_CONFIG/Indexer/20021213130740/NW\\_Reports.ddf](#)

© Copyright 1997-2002 edocs®, Inc. All Rights Reserved.  
edocs is Reg. U.S. Pat. & Tm. Off. [Privacy Policy](#)

9. Under Dynamic Web Views, next to XML, click 0 (Number of Auxiliary files). Publisher displays the Create a Version Set For XML screen:

**Create a version set for XML**

Application:

View Type: XML

View Name:

DDF File:

© Copyright 1997-2002 edocs®, Inc. All Rights Reserved.  
edocs is Reg. U.S. Pat. & Tm. Off. [Privacy Policy](#)

10. Select the application name from the drop-down list.
11. Enter `xmlDetail` for the View Name.
12. Browse and select a DDF file for the Indexer job. Enter the path to the same DDF entered for the Version Set for Indexer. For example, *EDCSbd/samples/NW\_Reports/NW\_Reports.ddf*.
13. Click **submit**. Publisher displays the Submission screen with details about the DDF file:
14. Close the Publisher window.
15. At the Create New Job screen in Command Center, click **Configure Job and Continue**. eaDirect displays the job configuration screen:



**Application: myReport Job: Indexer**

From this screen, all parameters of the selected job can be modified. To edit parameters, change the entries in the desired fields and click the Submit Changes and Schedule button. To Reset the fields or for Help click the appropriate button at the of the screen.

Submit Changes and Schedule Refresh Reset Help

**Task 1: Scanner**

Input File Path: /export/spare/EDCSbd/Input/myReport/

Input File Name: \*.\*

Output File Path: /export/spare/EDCSbd/Data/myReport/

**Task 2: Indexer**

DDF Path: /export/spare/EDCSbd/AppProfiles/myReport/DOC\_CONFIG/Indexer/200212131806/NW\_Reports.ddf

Doc Date: ----Today's Date----

Index Field List:

AmtDue	>>	----
CustName	>>	----
DueDate*	>>	----
StatementDate	<<	----

----

**Task 3: IXLoader**

16. For Task 2, select all the Index Fields. The fields that you select to be included in the Report output must also be defined in the RDX file.
17. When finished entering configuration parameters, click **Submit Changes and Schedule**. eaDirect asks “OK to submit this configuration?”
18. Click **OK**. eaDirect submits the job configuration parameters and displays the Schedule screen. You can specify the Indexer job schedule later.
19. Click **Main Console**.

## Creating and Configuring a Report Job

Before creating the Report job, you must set the field lengths in the DDF file for the fields that will be indexed by this job. If you fail to set the field lengths, the IXLoader task of the Reporting job will fail. See the *eaDirect User Guide* for information about editing field lengths.

### Creating and Configuring a Report Job:

1. Create a job for your application, select the Job Type **Report**.
2. Click Launch Publisher, and click Create. Under Batch Jobs, next to Report, click 0 (Number of Auxiliary files). Publisher displays the Create a Version Set For Report screen:

**Create a version set for Report**

Application:

View Name: Report

Report Description XML file:

Browse to find the sample report description XML, select it, and click Submit to publish the XML file as a version set. For example, *EDCSbd/samples/NW\_Reports/nwReportDescription.xml*.

Then close the Publisher browser window.

3. Submit the job, which brings up the job configuration screen.

### Application: myReport Job: Report

From this screen, all parameters of the selected job can be modified. To edit parameters, change the Changes and Schedule button. To Reset the fields or for Help click the appropriate button at the top of

Submit Changes and Schedule Refresh Reset Help

#### Task 1: StatementScanner

Index Volume Status: Accepted

Scan IVN Starting From (Number of Days): 7

Account Resolver: JNDI name of the Account Resolver

Account Attribute:

#### Task 2: ReportTask

Report Description XML File: /export/spare/EDCSbd/AppProfiles/myReport/DOC\_CONFIG/Report

XML View Name: XmlDetail

Report File Path: /export/spare/EDCSbd/ Input/

#### Task 3: ReportCollector

Output File Path: No user configuration necessary

### Task 1: StatementScanner

The StatementScanner task scans the XML view to find new data.

StatementScanner Task Configuration (Report job)	
Field	What to enter
Index Volume Status	Specifies that the job can proceed when a date appears in either the Date Accepted or Date Rejected column in the volumes table.

<b>StatementScanner Task Configuration (Report job)</b>	
<b>Field</b>	<b>What to enter</b>
	<p>Accepted</p> <p>(Default) Choose this option if you do not have custom verification/audit application or if you have one and you want to run the job only after a volume has been approved.</p> <p>If you choose this option, IVNScanner looks for a date in the Date Accepted column; if it contains a date, it proceeds to run the job.</p>
	<p>Rejected</p> <p>Use this setting to run the job even if a volume has been rejected in the custom verification application (use this option for testing purposes only).</p> <p>If you choose this option, IVNScanner looks for a date in the Date Rejected column; if it finds a date, it proceeds to run the job.</p>
Scan IVN starting from (number of days)	Specify how many previous days' volumes to scan for; IVNScanner selects any volumes indexed on or between the current date and the number of days ago you specify. (7 is the default)
Account Resolver	Specifies the Resolver used for user account information. For example, <i>edx/eaSample/ejb/CDAAccountResolver</i>
Account Attribute	Specifies the attribute that the statement scanner will use to identify users. Use UID for Report job.

### Task 2: ReportTask

The Report task uses the report description XML file specified here to extract data from the XML view named XMLDetail that was created when running the Indexer job. It then writes report files, one for each docId in the eight subdirectories created for the eight report types.

<b>ReportTask Configuration (Report job)</b>	
<b>Field</b>	<b>What to enter</b>
Report description XML	Contains the path to the last Report Description XML file you published.
XML View name	The name of the view that you published for this job when configuring the Indexer job. The default is XmlDetail.
Report file directory	The path to the directory where the Report job writes the reports. Directories are created for each report. eaDirect 3.4 creates eight directories, as described in the following table.

<b>Subdirectory Name</b>	<b>Description</b>
<applicationName>_r1	Top 20 most expensive calls
<applicationName>_r2	Top 20 longest calls
<applicationName>_r3	Top 20 most expensive called numbers
<applicationName>_r4	Top 20 most called numbers
<applicationName>_r5	Top 20 most called countries
<applicationName>_r6	Cost sum by call types
<applicationName>_r7	Cost sum by months
<applicationName>_r8	Cost sum by time periods

### **Task 3: Collector**

The report task generates one file for each docId in each of the report subdirectories. For each report type, the Collector task combines the reports for each docId into a single file.

No configuration is necessary for this task. Submit and schedule the job. Be sure you have enabled concurrency; see *Running Jobs Concurrently* for more information.



#### **Caution**

There must be at least one user enrolled to the application for the Report job to run successfully. Otherwise, the job will report a No-op.

## Indexing the Generated Report Data

The data generated by the Report job must be indexed so that the user can see it. You can use the sample DDFs with your ALF and HTML files to index the report data that was generated by the Report job. Create an application for each report type, and index the generated data so that it can be viewed with eaSample.

### To index generated report data for Top 20 most expensive calls:

1. Create a new application for <applicationName>\_r1, called **myReport\_r1**.
2. Create an Indexer job for that application. When publishing, use the following sample information:

Resource/Parameter	Value
View type	HTML
View name	HtmlDetail
DDF	/opt/EDCSBd/samples/reports/reports_ddf/r1.ddf
ALF	/opt/EDCSBd/samples/NW_Reports/NW_Reports.alf
HTML	/opt/EDCSBd/samples/NW_Report/NW_Reports.htm
Input File Path	/opt/EDCSBd/Input/myReport_r1
Input File Name	*.txt
Output File Path	Keep the default

3. Repeat the previous steps for the remaining report directories, using the DDF, ALF, and HTML for the appropriate report directory (for example, r2 for the second report).

## Testing Reporting

If you have used the sample DDFs and data files to install and configure the reporting feature, then you can use the following procedure to view the reports using the **reportSample** application.



**Caution**

---

If you have not already installed *ear-eaSample.ear*, then do that first. Refer to the *eaDirect Installation and Configuration Guide* for your platform for more information about installing *eaSample..*

---

1. Log on to the sample report application, using one of the accounts from the file that you indexed for the Report job. The URL is *http://host:port/eaSample&ddn=applicationName*. For example:  
`http://myHost:7001/eaSample?ddn=myReport`
2. Click on the Report link to view the reports.





## Overview of Concurrency

eaDirect 3.4 lets you configure the Scheduler to enable multiple instances of an application job to run in parallel. The Command Center lets you monitor and manage the individual job instances to keep your production environment running efficiently.

Until now, eaDirect ran all jobs sequentially, requiring one job instance to complete before another the job could start again. If an application job processed large or multiple input files, repeating the job sequentially may not have allowed enough time to complete the job before another input file appeared in the input directory.

Running jobs in parallel enables you to leverage machine power to process a large amount of data in less time. You can configure the maximum number of concurrent job instances to allow for each job: 5, 10, 15, or 20.

Concurrency is available with thread-safe jobs only. **PurgeApp** and **PurgeLogs** are not thread-safe and can run only one occurrence at a time.

Concurrency also makes it possible for the new Report job to process statements in parallel with the StatementScanner task. See “Multiple Statement Processing” on Page 43 for more information.

## Summary Job Information on the Main Console

For each eaDirect application, the Main Console lists each configured job type alphabetically. Although there can be multiple instances of an individual job for an application, the Main Console can display only one, so it chooses a representative job instance. eaDirect sorts all job instances first by status “ranking” and then by last run time in reverse chronological order. It selects the top-most instance from that list as the representative instance.

The Main Console displays the following information:

<b>Command Center Main Console</b>	
<b>Column</b>	<b>Description</b>
Application	Name of the eaDirect application.
Job Name	Name of the job.
Job Type	The purpose of the batch job: Indexer, Email Notification, Purge Apps, etc.
Last Run	Date and time the representative job instance ran.
Run Time	Elapsed time the representative job instance has been running in hours, minutes, and seconds.
Status	Current execution state of the representative job instance.
Next Run	Date and time the job is scheduled to run next. (This applies only to the job and not a particular instance.)
Action	Displays a button that lets you take action on that job. The Run Now button lets you run the job once immediately, overriding the scheduling parameters (except concurrency parameters). The Retry button lets you retry all failed instances of the job.

Jobs can have the following status, shown here in the order used for ranking purposes:

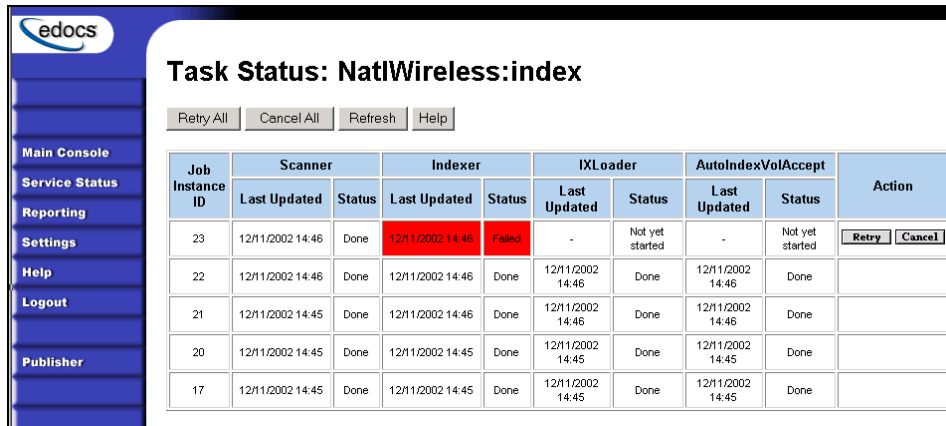
<b>Job Status (In Ranking Order)</b>	<b>Description</b>
Failed	Job failed
Processing	Job is currently executing
Reprocessing	Job is currently executing after a user manually selected it for reprocessing using the Retry or Retry All button
Reprocess	A user has manually selected the job for reprocessing using the Retry or Retry All button, but the job has not yet begun
No operation	Job/Task did nothing as resources were not ready yet, for example, if the Scanner task found no file in the input directory.
Done	Job has completed successfully
Canceled	Job run failed and was canceled
Not yet started	Job has not begun executing
Done, recurring	Job completed successfully and has been scheduled to run again, or the job has processed one data file and is looking at the input directory to see whether there are any more data files to process in this run
No operation, recurring	Previous job run resulted in a "No operation" status, but the job has been scheduled to run again
Canceled, recurring	Job was canceled and is now looking at the input directory to see whether there are any more data files to process in this run
Not scheduled	Job has not been scheduled to run

## Managing Individual Job Instances

You can monitor and manage individual job instances using the Command Center Task Status page.

## Running Jobs Concurrently

The Task Status page displays each instance of a job started during the most recent scheduled run in reverse chronological order (youngest first), along with the status of each task in the instance:



**Task Status: NatlWireless:index**

Retry All Cancel All Refresh Help

Job Instance ID	Scanner		Indexer		IXLoader		AutoIndexVolAccept		Action
	Last Updated	Status	Last Updated	Status	Last Updated	Status	Last Updated	Status	
23	12/11/2002 14:46	Done	12/11/2002 14:46	Failed	-	Not yet started	-	Not yet started	Retry Cancel
22	12/11/2002 14:46	Done	12/11/2002 14:46	Done	12/11/2002 14:46	Done	12/11/2002 14:46	Done	
21	12/11/2002 14:45	Done	12/11/2002 14:46	Done	12/11/2002 14:46	Done	12/11/2002 14:46	Done	
20	12/11/2002 14:45	Done	12/11/2002 14:45	Done	12/11/2002 14:45	Done	12/11/2002 14:45	Done	
17	12/11/2002 14:45	Done	12/11/2002 14:45	Done	12/11/2002 14:45	Done	12/11/2002 14:45	Done	

The Task Status page identifies each job instance by a Job Instance ID. The Task Status page displays the following information:

<b>Command Center Task Status Page</b>	
<b>Column</b>	<b>Description</b>
Job Instance ID	A number uniquely identifying each job instance.
Last Updated	The time the task status last updated.
Status	Current execution state of the task. Task Status can be: Processing, Failed, Reprocessing, Reprocess, No operation, Canceled, Not yet started, or Done.
Action	Displays a button that lets you take action on that job instance or on all instances. The Retry button lets you retry that instance; the Retry All button lets you retry all failed instances of the job. The Cancel button lets you cancel that instance; the Cancel All button lets you cancel all failed instances of the job.

## Number of Rows On the Task Status Page

The Task Status page displays up to the last  $N$  job instances that have Done, Canceled, or No operation status (where  $N$  is the maximum number of concurrent instances allowed for the job), plus any instances in Processing, Failed, Reprocessing, or Reprocess status.

If you are not using concurrency ( $N=1$ ), the Task Status page shows up to five rows of job instances in Done, Canceled, or No operation status, plus any instances in Processing, Failed, Reprocessing, or Reprocess status.

When a scheduled run completes, the completed rows remain in view on the Task Status page until a new schedule begins. At this point, the Task Status page begins displaying the instances generated by the new schedule instead. The only exception is that any instances from the previous schedule still in Processing, Failed, Reprocessing, or Reprocess states remain even if a new schedule has begun. eaDirect removes those instances from the Task Status page once processing is complete, or in the case of a failed instance, once you cancel or retry it successfully.

Schedules can overlap if a second schedule begins before the current run completes. Another scheduled run can begin only if:

- The first run is not using the maximum number of instances (if enough “resource” is available). For example, if the first run has 3 instances in Processing and the maximum allowed is 10, the next run can start up to 7 new instances.
- No job instances in the first are in the Failed state.

Overlapping schedules mean that instances from both schedules could appear on the Task Status page. You can tell from the Last Updated field to which schedule the instance belongs.

The number of rows that appear on the Task Status page at any given time depends on the point of progress of the job plus:

- Whether you have enabled concurrency for the job (if the maximum number of instances specified in the schedule is >1).
- The maximum number of concurrent jobs you allow. This number is also the maximum number of Done, Canceled, or No operation jobs that can appear on the Task Status. If you are not using concurrency, the Task Status shows a maximum of 5 job instances in Done, Canceled, or No operation.
- For jobs that scan for an input file, such as Indexer, the number of input files placed in the input directory.

For jobs that process multiple statements in parallel with the **StatementScanner** task, such as the Report job, the number of statements to process up to the maximum number of instances.

- Whether the job schedule overlaps due to a long lasting run.

**Examples: Task Status for the Indexer Job**

<b>Concurrency Enabled?</b>	<b>Number of Input Files</b>	<b>Maximum Instances</b>	<b>Number of Rows that Can Appear on the Task Status Page</b>
No	3	1	1, 2, or 3. Each job instance must complete before another can appear. If all have been completed, 3 Done instances will show; by default the Task Status page can show up to 5 completed instances when the job runs sequentially (not concurrently).
Yes	3	10	1, 2, or 3. Three job instances can run concurrently and appear at once on the Task Status screen.  (Up to 10 rows could appear at once if the job schedules overlap and more input files subsequently appear in the input directory during the second scheduled run.)
Yes	10	3	1-6. As few as one row with status in Processing or, if processing is complete, 3 rows all in Done status can appear.  For example, at some point you could see 6 rows; the first instance might be in Processing, the second, third, and fourth in Done, the fifth in Failed, and the sixth in Processing, with the rest not started and therefore not shown.

**Canceling and Retrying Failed Jobs**

You can use the Main Console to cancel or retry a job, and the Task Status page to retry or cancel a failed instance of a job.

If one instance fails, other instances that have started continue to completion, but eaDirect does not start any new instances.

Retry running a failed job or job instance if you want to start it from the point where it failed. If you want to restart a job or instances of a job, cancel and run it again.

If the task has not been started, the Last Update field shows “-“ and Status shows Not Yet Started.

**To retry a *failed job* before its next scheduled run time:**

- On the Main Console, click the **Retry** button for the failed job. Or, on the Task Status page, click **Retry All**, which retries all failed instances of the job.

eaDirect immediately restarts all failed instances at the task that failed in each instance, and changes the job status from **Failed** to **Reprocess**.

**To retry a *failed job instance*:**

- On the Task Status page, click the **Retry** button in the Action section next to the failed instance.

eaDirect immediately restarts the failed job instance at the task that failed and changes the instance status from **Failed** to **Reprocess**.

**To cancel all instances of a *failed job*:**

- On the Task Status page, click the **Cancel All** button.

eaDirect cancels all failed instances of the job and changes the job status to Canceled, which remains until the next time the job is scheduled to run again.



**To cancel a *failed job instance*:**

- On the Task Status page, click **Cancel** in the Action section next to the failed instance.

eaDirect cancels the failed job instance and changes the job instance status to **Canceled**, which remains until the next time the job is scheduled to run again.

## Configuring and Scheduling Concurrent Jobs

By default, eaDirect runs a single thread of each application job. To enable a job to run multiple threads, you must configure the job schedule and specify the maximum number of instances to allow at once: 5, 10, 15, or 20.

**To configure a job to enable multiple instances to run:**

1. At the eaDirect Command Center, locate the application and job you want to schedule to run multiple instances of, and click its status in the Next Run column. The Schedule screen appears. (If you just completed configuring a job, the Schedule screen appears automatically.)

The screenshot shows the 'Schedule Job: NatlWireless:Index' interface. On the left is a navigation menu with items: Main Console, Service Status, Reporting, Settings, Help, Logout, Publisher, and www.edocs.com. The main content area has the following sections:

- Schedule Date:** Start Date: [ ] [Popup Calendar](#)
- Schedule Window:** Start Time: 12 am : 00
- Recurring:**
  - Try once
  - Try every 5 minutes until 12 am : 00
  - Do not repeat this event
  - Repeat Every Day
  - Repeat on day 1 of the month every month
  - Forever
  - Until [ ] [Popup Calendar](#)
- Concurrency:**
  - Do not run multiple job instances, only one at a time
  - Run maximum number 5 of concurrent job instances

Buttons at the bottom: Save Schedule, Clear Schedule, Run Now, Help.

2. Under the Concurrency section, select the **Run maximum number of (5) job concurrent instances** option.
3. To change the maximum number of instances to allow, click the  dropdown box and choose 5, 10, 15, or 20.
4. Specify any other changes you need to make to the schedule parameters, if any. To clear the screen to reenter all parameters, click **Clear Schedule**.
5. When finished setting the schedule, click **save schedule**. eaDirect saves this schedule and adds the job to the production queue, overriding any scheduling parameters you set for a single execution of the job.

**To configure a job to run a single instance:**

- Follow the procedure above to configure multiple job instances, except choose the **Do not run multiple job instances, only one at a time** option.

**Tip**

Click the Run Now button on the Main Console to run just one instance of a job.

---

## Multiple Statement Processing

eaDirect introduces parallel statement processing with the new **StatementScanner** task in the new **Report** job. StatementScanner automatically processes multiple statements from an input file at the same time. eaDirect creates an individual Report job instance for each statement in the input file it processes.

For example, if you have an input file with 1,000 statements whose accounts are resolved as valid and you set the maximum concurrency for this job at 20, scheduler can start up to 20 job instances to process 20 statements simultaneously.

You can manage the multiple instances of any job that has the StatementScanner task in much the same way you manage concurrent instances generated by other job types. See the information in this chapter about managing job instances using the Main Console and the Task Status page. Also, see the chapter about the new Report feature, including the new Report job and StatementScanner task.



## Other New Features in eaDirect 3.4

# 4

### Defining Extended Functions For Group Fields

Use Composer on group fields to create field elements that sort, filter, page, combine, calculate arithmetic, and analyze business conditions.

For more information on sorting, filtering, paging, and calculating on field elements in the Composer, see the *eaDirect User Guide*.

### Positioning Starting Columns For DB Extract Files

When using DefTool to create a new DDF for a DB Extract input file, you can now specify a starting column position if the page delimiter is a string.

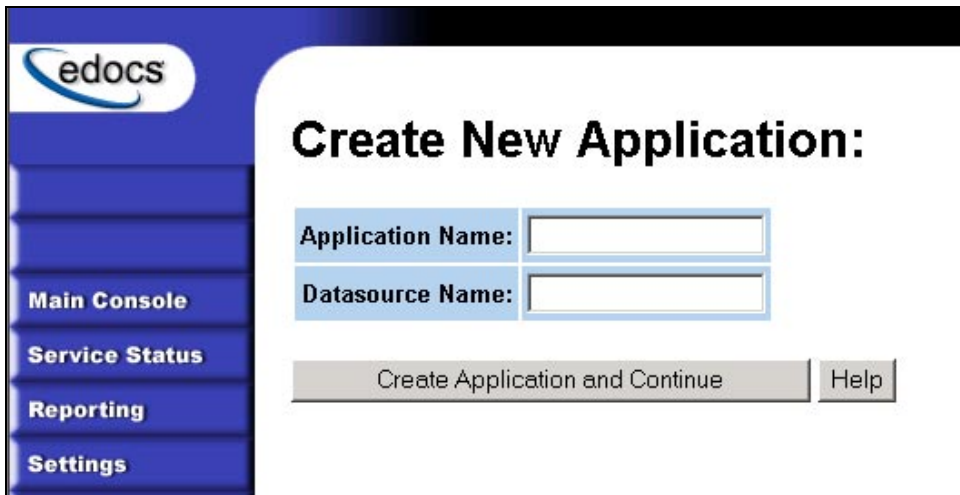
For more information on positioning columns and pages in a DDF, see the *eaDirect User Guide*.

### DDN to Datasource Mapping

Specify a custom datasource EJB for each eaDirect application (DDN) directly in the Command Center, without hand-editing deployment descriptors and redeploying your web application. A custom datasource can retrieve, for example, live data from an external database or archival data from offline storage. In some cases, customizing the datasource can also improve performance and save disk space. For information on developing a custom datasource EJB, please consult your edocs Professional Services representative.

In this context, a **datasource** refers to an EJB in your web application (EAR file) that specifies summary information and location of your document data. When your developers build your custom web application, the environment entries in its deployment descriptors map the JNDI name of this datasource EJB to your DDN, or eaDirect application.

eaDirect 3.4 allows you to change this JNDI mapping at the DDN level, without modifying, repackaging, and redeploying your web application. When you create a new eaDirect application in the Command Center, you must now specify the name of the related datasource EJB.



This gives you the flexibility to define different datasource EJBs for different DDNs. Each web application shipped with eaDirect has a default datasource. If you do not need to customize your datasource, simply enter the default datasource name for eaDirect, `edx/ejb/EdocsDataSource`.

Web Application	Datasource Name (default EJB)
eaDirect	<code>edx/ejb/EdocsDataSource</code>
eaSample	<code>edx/eaSample/ejb/EdocsDataSource</code>
eaTraining	<code>edx/eaTraining/ejb/EdocsDataSource</code>
hierarchySample	<code>edx/hierarchy/ejb/EdocsDataSource</code>

## About Native XML Input

eaDirect can extract and publish your data as XML using schemas, stylesheets, and other XML design files. Native XML input enhances end-to-end XML support in the eaSuite by accepting your data as well-formed XML, without the need for a Data Definition File (DDF). This scalable, persistent edocs solution lets you manage large volumes of XML data in real time, while saving the system costs of processing and composing raw XML.

eaDirect has two new Command Center batch jobs and a new type of dynamic web view designed expressly to accept XML input.

The **XML Loader** job processes an XML input file, much as the Indexer job handles AFP and other print files. However, unlike the Indexer job, the XML Loader job does not extract only selected (*indexed*) data for storage in the eaDirect database. In the XML model, eaDirect applies attribute tags to the entire input file, and the modified XML file is then stored in the XML repository for “extraction” at runtime with XSL stylesheets that process only the tagged data defined in the stylesheet.

The **XML Email Notification** job processes batch email from XML data, much as the Email Notification job handles batch email for AFP and other print files. An additional job setting allows you to specify a mail auditor. Instead of a DDF, ALF, and HTML templates, the XML version set for XML email notification takes an XSL stylesheet, and creates an XML Services (XS) web view type named **XENotification**.

The content and structure of the input XML is preserved while tagging the input XML data with attribute metadata. Input data is not removed or modified by applying attributes, and the modified data is stored persistently in the XML repository.

The XS (XML Services) web view publishes XML data much like an HTML view. Instead of a DDF, ALF, and HTML templates, the XML version set for this job takes an XSL stylesheet.

This chapter describes the production steps needed to process XML input.

## Overview of XML Input Workflow

1. Create an XML schema for your native XML input file.
2. Create XSL style sheets for each view to be presented.
3. Work with edocs Professional Services to edit your schema to include edocs attributes for annotation, and to customize the data source in your web application as needed to use XML.
4. Create an eaDirect application for your XML jobs.
5. Create an XML Loader job.
6. Publish at least one XML Loader batch view.
7. Publish one or more XS (XML Services) web views for XML Loader.
8. Configure and schedule your XML Loader job
9. Create an XML Email Notification job.
10. Publish a batch view (**XENotification**) for XML Email Notification.
11. Configure and schedule your XML Email Notification job.

The topics below describe these steps in more detail.



## About XML Input Files for eaDirect

You will need four types of XML files for using XML input with eaDirect:

- **Data (\*.xml)**: one or more valid, well-formed XML file(s), which may consist of multiple XML files each containing multiple statements.
- **Schema (\*.xsd)**: a valid W3C XML schema file of your XML data file(s). DTDs must be converted into a schema for use with eaDirect.

---

**Tip**

While it is not strictly necessary, it is good practice to name each schema file with the name of the related eaDirect application.

---

- **Attribute (edx.xsd)**: a special type of schema file defining edocs-specific variables that identify fields in your document with metatags, for extraction at runtime.

---

**Tip**

Attribute files work closely with your input data. Consult edocs Professional Services to develop a custom edx.xsd file for your dataset.

---

- **Stylesheet (\*.xsl)**: one or more valid, well-formed XSL stylesheets, one for each web view you wish to publish.

For information on how to prepare your XML data files, schemas, attributes and stylesheets for eaDirect, please consult your edocs Professional Services representative.

## Enabling Your Web Application for XML Input

Your web application will need to point its deployment descriptors to a custom EJB defining an XML data source for eaDirect data. For information on customizing your web application, please consult your edocs Professional Services representative.

## Creating and Configuring an XML Loader job

You must create and configure an XML Loader job to use XML data for live retrieval with your application. This job requires you to:

- Publish the schema and attribute files created for your XML Loader job
- Specify configuration settings for the three production tasks that run sequentially as part of the XML Loader job: Scanner, XMLLoader, and AutoIndexVolAccept

Review all the task and field configuration settings for this job to determine which options to use in your application.

Each time you run an XML Loader job, it looks for multiple data files in the input directory and processes them one at a time.

### To create and configure an XML Loader job:

1. Create an XML schema for your native XML input file.
2. Create an eaDirect application for your XML input file, in this example **xmltest**.

3. The **Create New Job** screen appears automatically after you create a new application. Otherwise, click the application name on the Main Console, then click the **Add New Job** button. eaDirect displays the **Create New Job** screen:

**edocs**

## Create New Job: xmltest

When creating a job for an application that has just been created, you will need to publish the ALF, DDF and associated HTML template files. For adding additional jobs to an existing application, publishing the files is often not necessary, unless you wish to modify or create new versions of these files.

- 1 Name new job and select a job type.
- 2 Publish
- 3 Configure

**Job Name:** xmlloader

**Job Type:** XML Loader

Detail Extractor  
Email Notification  
HTML Output  
Indexer  
Purge App  
Purge Logs  
XML Email Notification  
XML Loader  
XML Output

Launch Publisher

Configure Job and Continue Help

www.edocs.com

4. Enter a meaningful name for the XML Loader job, in this example **xmlloader**. The job name may contain alphanumeric characters and underscores, but no spaces. The XML Loader job name may not start with a numeric.
5. Select the XML Loader job type from the drop-down menu.
6. Click Launch Publisher.

- Click **Create**. Publisher displays the **Select a Version Set Type** screen:

**edocs**

**Select a Version Set Type**

**Dynamic Web Views**

Job Type	Number of Auxiliary Files
HTML	0   1   2   3   <a href="#">more</a>
XS	0   1   2   3   <a href="#">more</a>
CSV	0
XML	0
CHART	0
XSLT	0
XMLQuery	0

**Batch Jobs**

Job Type	Number of Auxiliary Files
Detail Extractor	0
Email Notification	0   1   2   3   <a href="#">more</a>
HTML Output	0   1   2   3   <a href="#">more</a>
Indexer	0
XML Email Notification	0
XML Loader	0
XML Output	0

[www.edocs.com](#)

- Under Batch Jobs, next to XML Loader, click 0 (Number of Auxiliary files). Publisher displays the **Create a Version Set For XML Loader** screen:

The screenshot shows a web interface with a blue sidebar on the left containing the 'edocs' logo and a menu with buttons for 'Browse', 'Create', 'Fetch', and 'Delete'. The main content area is titled 'Create a version set for XML Loader'. It contains the following form elements:

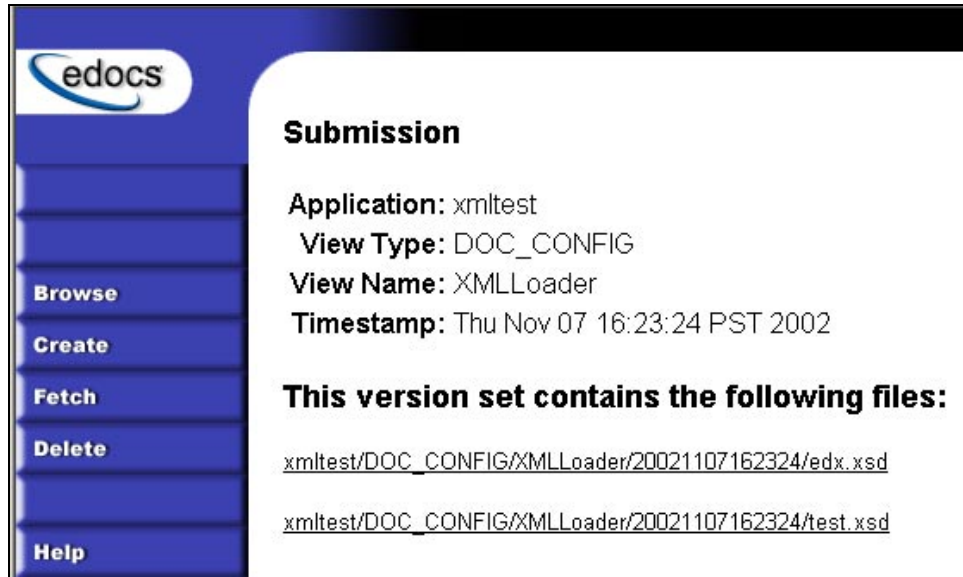
- Application:** A drop-down menu with 'xmltest' selected.
- View Name:** A text input field containing 'XMLLoader'.
- Schema File:** A text input field containing 'test.xsd' and a 'Browse...' button to its right.
- Attribute File:** A text input field containing 'edx.xsd' and a 'Browse...' button to its right.
- At the bottom of the form are three buttons: 'Submit', 'Clear', and 'Help'.

9. Select the application name from the drop-down list.
10. Browse and select the XML Schema file (for example, `test.xsd`) and Attributes file (`edx.xsd`) for the XML Loader job.



Make sure you enter the schema XSD file in the Schema field, and the attribute XSD file in the Attribute field. If you reverse these files, the job will run, but will not display correctly.

11. Click **Submit**. Publisher displays the Submission screen with details about the schema and attribute files:



12. Close the Publisher window.
13. At the **Create New Job** screen in Command Center, click **Configure Job and Continue**. eaDirect displays the job configuration screen:

**Application: xml Job: xmlloader**

From this screen, all parameters of the selected job can be modified. To edit parameters, change the entries in the desired fields and click the Submit Changes and Schedule button. To Reset the fields or for Help click the appropriate button at the top of the screen.

Submit Changes and Schedule Refresh Reset Help

**Task 1: Scanner**

Input File Path:

Input File Name:

Output File Path:

**Task 2: XMLLoader**

Schema File Path:

Attribute File Path:

**Task 3: XIRLoader**

Skip Rows:

Split Size:

Optional Field Count:

Load Method:

**Task 4: AutoIndexVolAccept**

Action on Index Volume:

14. Specify the configuration parameters for each of the tasks that run as part of the XML Loader job. Carefully read the descriptions of each task and field (in the section below) to choose the appropriate values for your application.
15. When finished entering configuration parameters, click **Submit Changes and Schedule**. eaDirect asks "OK to submit this configuration?"
16. Click **OK**. eaDirect submits the job configuration parameters and displays the Schedule screen. You can specify the XML Loader job schedule later.
17. Click **Main Console**.

## Task 1: Scanner

The Scanner task scans the input directory for new data input files. When it finds a new data file, it moves the file to the output directory and renames it, adding a timestamp (YYYYMMDDHHMMSS\_filename.ext). When the Scanner finds multiple files, it processes them one at a time.

**Tip**

For the Input File Name in an XML Loader job, edocs recommends entering \*.xml to search for all XML files in the Input directory.

For more information, see the *eaDirect Production Guide*.

## Task 2: XMLLoader

The XMLLoader task uses the data file (from the data directory) and the published XMLLoader XML schema and attribute files (from the job configuration), and annotates the XML data fields to be indexed in a binary file in the edocs XML repository. It also creates an XML Intermediate Representation (.XIR) file.

Field	What to enter/select
Schema File Path	(Not an editable field.) The directory path and name of the XML Schema file. This is the XML Schema you specify when publishing the XMLLoader version set.
Attribute File Path	(Not an editable field.) The directory path and name of the XML attributes file (edx.xsd). This is the Attribute File you specify when publishing the XMLLoader version set.

## Task 3: XIR Loader

The **XIRLoader** task uses the database loader to load data from the .XIR file to a database table using the script information in the .CMD file. It creates a row in the index table for each primary key. XIRLoader also creates the .LOG, .CTL, and .CMD files.

XIRLoader Task Input:	XIRLoader Task Output:
<ul style="list-style-type: none"> <li>.XIR file</li> <li>Settings from the job configuration</li> </ul>	<ul style="list-style-type: none"> <li>Rows added to index tables</li> <li>.LOG, .CTL, and .CMD files</li> </ul>

The XIRLoader task takes the same user inputs as the IXLoader task of the Indexer job. For more information, see the *eaDirect Production Guide*.



## Task 4: AutoIndexVolAccept

The AutoIndexVolAccept task determines whether the system can make the indexed data available for immediate user access or whether it must wait for you to approve the data. This task is primarily intended for eaDirect applications using a customized verification process.

For more information, see the *eaDirect Production Guide*.

## Creating and Configuring an XML Email Notification Job

You create an XML Email Notification job to create and send an email notification message to enrolled users whose data is input as XML.

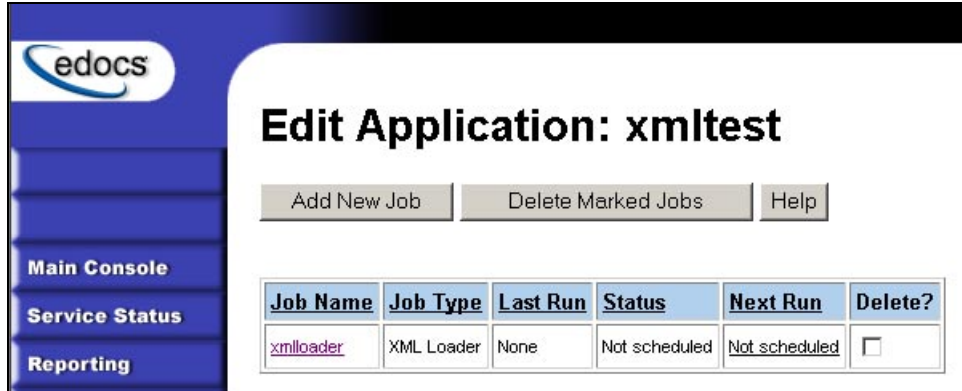
Creating and configuring an XML Email Notification job requires you to:

- Publish the XSL file created expressly for your XML Email Notification job.
- Specify configuration settings for the two production tasks that run sequentially as part of the job: **IVNScanner** and **XMLMailNotification**

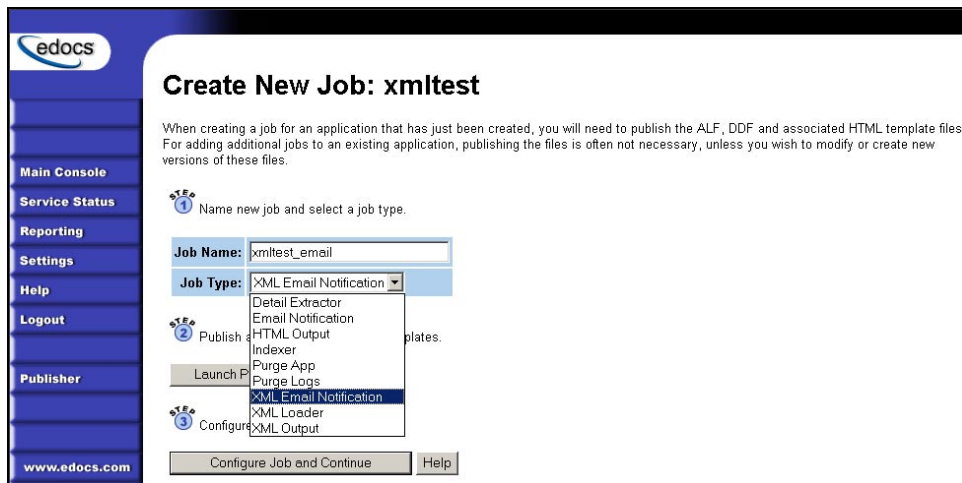
Review all the task and field configuration settings for this job to determine which options to use in your application.

### To create and configure an XML Email Notification job:

1. On the Main Console, click the application name in the table. The **Edit Application** screen appears.



2. Click Add New Job. eaDirect displays the **Create New Job** screen:



3. Enter a meaningful name for the job. The job name must start with an alpha character. The rest of the characters can be alphanumeric and can contain underscores, but no spaces.
4. Select the **XML Email Notification** job type from the drop-down menu.
5. Click **Launch Publisher**.

**Tip**

XML Email Notification has its own XS View Name, XENotification. This name is not modifiable. To browse for an existing XML Email Notification job, open the Publisher's Browse window and use the drop-down menus to select **View Type>XS** and **View Name>XENotification**.

6. Click **Create**. Publisher displays the **Select a Version Set Type** screen:

**edocs**

**Select a Version Set Type**

**Dynamic Web Views**

Job Type	Number of Auxiliary Files
HTML	0   1   2   3   <a href="#">more</a>
XS	0   1   2   3   <a href="#">more</a>
CSV	0
XML	0
CHART	0
XSLT	0
XMLQuery	0

**Batch Jobs**

Job Type	Number of Auxiliary Files
Detail Extractor	0
Email Notification	0   1   2   3   <a href="#">more</a>
HTML Output	0   1   2   3   <a href="#">more</a>
Indexer	0
XML Email Notification	0
XML Loader	0
XML Output	0

**Browse**

**Create**

**Fetch**

**Delete**

**Help**

[www.edocs.com](http://www.edocs.com)

7. Under **Batch Jobs**, next to **XML Email Notification**, click 0 (Number of Auxiliary files). Publisher displays the **Create a Version Set for XML Email Notification** screen:

The screenshot shows the edocs application interface. On the left is a blue sidebar with the edocs logo and buttons for 'Browse', 'Create', and 'Fetch'. The main content area has a white background with a blue header. The title is 'Create a version set for XML Email Notification'. Below the title are three input fields: 'Application:' with a dropdown menu showing 'xmltest', 'View Name:' with the text 'XENotification', and 'XSL File:' with a text box containing 'tiveXML\samples\testApp\test.xml' and a 'Browse...' button. At the bottom of the form are three buttons: 'Submit', 'Clear', and 'Help'.

8. Select the application name from the drop-down list. Browse to select the XSL file for this version set.
9. Click Submit. Publisher displays the Submission screen showing the version set you just published:

The screenshot shows the edocs application interface. On the left is a blue sidebar with the edocs logo and buttons for 'Browse', 'Create', 'Fetch', and 'Delete'. The main content area has a white background with a blue header. The title is 'Submission'. Below the title are four lines of text: 'Application: xmltest', 'View Type: DOC\_CONFIG', 'View Name: XENotification', and 'Timestamp: Thu Nov 07 16:55:00 PST 2002'. Below this is a bold heading 'This version set contains the following files:' followed by a single line of text: '[xmltest/DOC\\_CONFIG/XENotification/20021107165500/test.xml](#)'.

10. Close the Publisher window.
11. At the **Create New Job** screen in Command Center, click **Configure Job and Continue**. eaDirect displays the XML Email Notification job configuration screen.

**edocs**

**Application: xmltest Job: xmltest\_email**

From this screen, all parameters of the selected job can be modified. To edit parameters, change the entries in the desired fields and click the Submit Changes and Schedule button. To Reset the fields or for Help click the appropriate button at the top of the screen.

Submit Changes and Schedule Refresh Reset Help

**Task 1: IVNScanner**

Index Volume Status: Accepted

Scan Starting From (Number of Days): 7

**Task 2: XMLMailNotificationTask**

Base URL: http://Dino/eaSample/User?app=UserMain&jsp=/user/jsp/Detail.jsp

SMTP Hosts: SMTP host names (comma separated)

Return Address: Return email address

Subject Text: Email subject text

Administrator's Addresses: Administrator email addresses (comma separated)

Administrator Subject Text: Email Notification Error

Max Number of Retries: 3

Retry Period (min): 60

Enroll Model: JNDI name of the Account Resolver

Access Type:

Auditor Model: JNDI name of Mail Auditor

Main Console  
Service Status  
Reporting  
Settings  
Help  
Logout  
Publisher  
www.edocs.com

12. Specify the configuration parameters for each XML Email Notification task. Carefully read the descriptions of each task and field to choose the values appropriate for your application and job.
13. When finished entering configuration parameters, click Submit Changes and Schedule. eaDirect submits the job configuration parameters and displays the Schedule screen. You can schedule the XML Email Notification job later.
14. Click Main Console.

## Task 1: IVNScanner

The IVNScanner task determines whether index data has been verified before creating and sending email to enrolled customers. This task is primarily intended for applications with a customized verification/audit application.

For more information, see the *eaDirect Production Guide*.

## Task 2: XMLMailNotification

The XMLMailNotification task builds and sends email notifications *from XML data*. It uses information from the job configuration settings, the most recently published XML email version set, schema references, the data file, and the email addresses to generate email.

For information on configuring email notification, see the settings for the Email Notification job in the *eaDirect Production Guide*.

Since eaDirect 3.1, email notification tasks have one additional setting, **Auditor Model**. In this field, you specify the JNDI name of the Mail Auditor your system uses. Contact your system administrator to determine whether custom mail auditing has been implemented and if so, to obtain the appropriate JNDI name of your mail auditing client.

For more information on customizing your solution to audit e-mail notification, contact your edocs Professional Services representative.

## Sub-Account Indexing

### About Sub-Account Indexing

Sub-account indexing enables you to:

- Quickly access sub-account data in a data input file for presentment in a view. (DefTool is the eaDirect GUI application you use to define the data extraction rules for your input file.)
- Include sub-account information in a history list, or "hit list."

For example, you could define each section of cell phone detail for a large B2B telecommunications statement as a sub-account, speeding Web access to line detail on request.

To enable sub-account indexing, you must define the sub-account data area of an input file as a group, then define a group field on a piece of data that uniquely identifies the sub-account (or sub-document), called a subkey. Just as a primary key identifies a document, the subkey identifies each sub-document group.

You can also define additional group fields for indexing and presentment. eaDirect indexes the group fields that belong to the sub-account group as part of the regular Indexer production job, storing exact offsets and other information about the start and end of each sub-account in the database. This index data enables you to include sub-accounts in a history list and to click a link to quickly access sub-account data in a view (without having to scan the entire statement to find the group information).

You must create a separate view (DDF/ALF pair) for presenting sub-account information to users. You can also define more than one sub-account for an application; create one indexing DDF to use for all the sub-accounts and a separate ALF for each sub-account view.

The larger document that a sub-document belongs to is called the root, or parent, document. One parent document can consist of many instances of one or more sub-documents (sometimes referred to as the "children" of the parent document).

The following section describes the general process you must follow to set up sub-document indexing in your application.

## Sample Application For Sub-Account Indexing

eaDirect 3.2 includes a new sample application view to demonstrate sub-document indexing. The files are:

```
EDCSbd\samples\NW_Subdocument\NW_Subdocument.ddf
```

```
EDCSbd\samples\NW_Subdocument\NW_Subdocument.alf
```

Use the NationalWireless.txt data file with this application.

The **NW\_Subdocument** application contains two eaDirect views, **NW\_Subdocument** and **NW\_Subdocument\_Detail**. The **NW\_Subdocument** view contains account level data such as the primary field. The **NW\_Subdocument\_Detail** view is the Sub-Account view.

This new eaDirect sample application may be used with both the eaTraining and the hierarchySample web applications. These web applications have also been extended to support Sub-Account views. Two Dynamic HTML views are required to view Account and Sub-Account documents using eaTraining. The Account or Primary Dynamic HTML view must be named **HtmlDetail**. Sub-Account views should be named **SubDetail<n>**, where **<n>** equals the level of the Sub-Document.

## Setting Up Sub-Account Indexing

This section describes the general process required to implement sub-document indexing in a new or existing eaDirect application; each step is described in more detail in separate sections in this chapter. All of these instructions are supplemental to the *eaDirect User's Guide*, which contains complete instructions on using DefTool and Composer to define an application.



You must tailor these procedures to the particular data in your input file and the views you need to present. Include these steps in the overall plan, design, and implementation of your eaDirect application.

Note that the procedure to implement sub-document indexing in an existing eaDirect application is different from creating the feature in a new application. For an existing eaDirect application, you can use your existing DDF, however you must create a separate ALF for presenting the sub-document view.

### To implement sub-account indexing in a new eaDirect application:

1. Using DefTool, create a DDF specifically to index and extract data for the sub-account view. See the eaDirect User's Guide for details about using DefTool to create a new DDF and define extraction rules.
2. Define the doc style and the **primary key** in the indexing DDF, along with any other **pure fields** you need to index for your application. (You must index fields to make them available for inclusion in the history list.)
3. In this DDF, define the sub-document data as a **group**. (This includes creating group markers, creating the group, and assigning the group markers to a group.)
4. Identify a piece of data that uniquely identifies each sub-account in the data input file, such as sub-account number, and define this as the subkey group field for the sub-account. The subkey must be located within the sub-account group.

---

**Tip**

eaDirect uses the subkey to locate the requested sub-document during dynamic retrieval. When the Indexer job indexes the subkey, it stores the exact offsets to each sub-document. By storing offsets for each sub-document, eaDirect has the information it needs to directly access any sub-document in the input file, eliminating the need to search the entire data file to identify the sub-document boundary.

---

5. Define any **group fields** you want to index, such as the name or other information associated with the sub-account number. Define any group fields that need to be indexed within the sub-account parent group.

6. In Composer, create a new Application Logic File (ALF), selecting the "Use sub-document" option to identify it as a sub-document view. Map the tables and fields to the ALF default or parent template, which is essentially the group template. (Composer is the eaDirect GUI application you use to define the presentation rules for displaying statements on the Web.)

See "Creating an ALF for the Sub-Account View in Composer" in this chapter for details, along with the comprehensive instructions on using Composer in the *eaDirect User's Guide*.

7. When you configure the Indexer job using the eaDirect Command Center, publish the indexing DDF with the Indexer job. eaDirect automatically indexes all group fields defined in the parent group of a sub-account in the DDF. See the *eaDirect Production Guide* for details about fields you need to index in your application and other Indexer job configuration details.

#### **To add sub-account indexing to an existing eaDirect application:**



1. Make sure that you have defined the sub-account data as a group in your existing DDF.
2. Identify a piece of data that uniquely identifies the sub-account, such as a sub-account number. Right-click on the group name in the Definition Tree and select **Edit** from the right-click menu. On the Group Properties dialog, click **Create**. Define the field you want to use as a subkey.
3. On the Group Properties dialog, click **Create** again to define any additional group fields you want to index and/or include in the view, such as the name associated with the sub-account number. Define any group fields that need to be indexed within the sub-account parent group.
4. In Composer, you must define a new ALF for presenting the view of sub-account data, selecting the "Use sub-document" option to identify it as a sub-document view. Drag and drop (map) the tables and fields to the ALF default or parent template, which is essentially the group template. See "Creating an ALF for the Sub-Account View in Composer" for details.
5. In the Command Center, publish the updated DDF.
6. Run the Indexer job. eaDirect automatically indexes all group fields defined in the parent group of the sub-account in the DDF.

## Creating a DDF for Indexing and Sub-Account Presentment

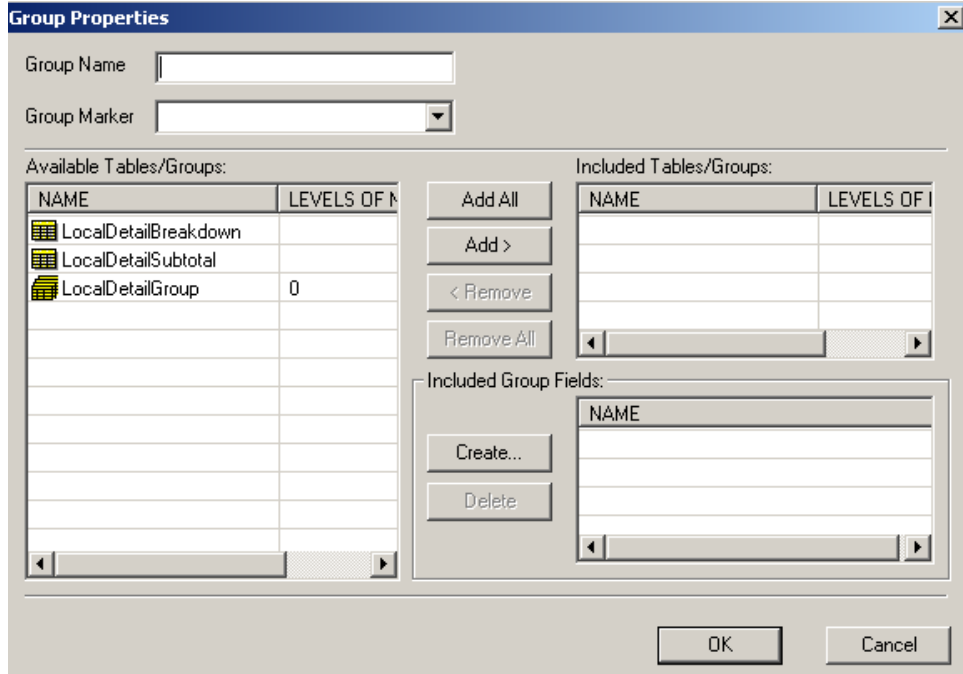
Follow the procedures in this section if you are creating a new eaDirect application and need to implement sub-account indexing. In the DDF you create for sub-account indexing, you must define all the fields that you need to index for the application.

At each step using DefTool or Composer, refer to the *eaDirect User's Guide* for complete instructions; these are too detailed to include here.

### To create a new DDF for sub-account indexing in DefTool:

1. In DefTool, select **New** from the File menu. DefTool begins the Data Configuration Wizard. Complete the wizard and follow the procedures described in the *eaDirect User's Guide* for creating a document style, primary key, and any additional page styles.
2. Click the New Marker icon  and define the start and end markers for the sub-account group. You define the start and end markers for the group indicating the group offsets - the exact location in the data input file where eaDirect should begin searching for the group data. Once you define a group, it appears in the Definition Tree on the left of the DefTool window for the DDF.
3. Click the Group icon  to display the Group Properties dialog. Define the properties of the sub-account group. Specify the group name, associated markers, and select the tables in the group.

Sub-Account Indexing



4. Create a subkey for the sub-documents: identify a piece of data in the data input file that uniquely identifies each sub-account. The subkey and any anchor you specify must be between the group's start and end markers.

5. On the Group Properties dialog, click **Create**. Use the left mouse button to drag the crosshair selector box around the data you want to use for the subkey group field. At the Select Your Choice dialog, select Field and click **OK**. DefTool displays the **Group Field Properties** dialog:

**Group Field Properties**

Field Information

Field Name:  New...

Pattern: 703 605-6900 Reg...

Select Characters >> Save Pattern... Select Pattern... Test

Page Style Name :

Window Coordinates

Start Row: 6 Start Column: 18

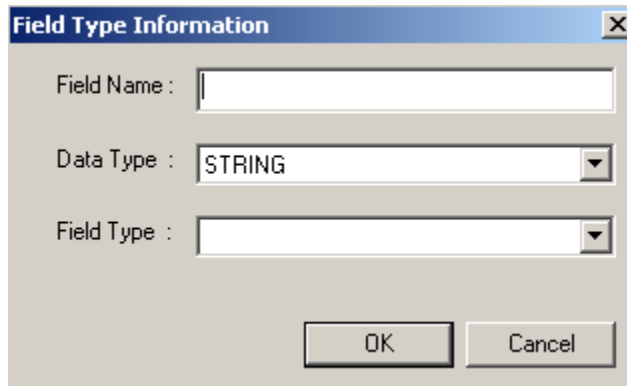
End Row: 7 End Column: 31

Force Fixed Length

NOTE: Field length is used to set the database field length for indexing.  
An incorrect value may cause problems during indexing.


Formatting Specifications... OK Cancel

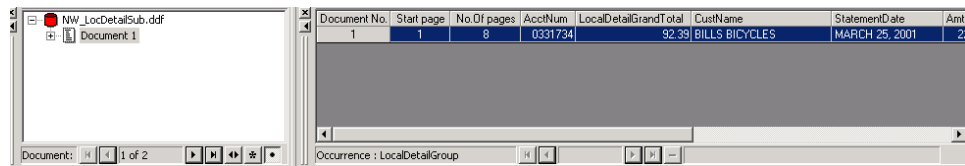
6. Click **New**. DefTool displays the Field Type Information dialog. Specify a name for the subkey field, select the data type, and select Sub key from the Field Type drop-down list.



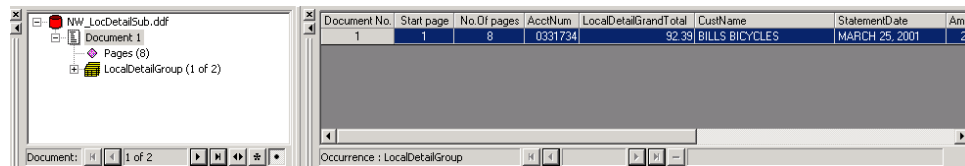
7. Click **OK**. Continue to define extraction rules (properties) for each subkey group field, including the page style to associate with the group field. Click **OK** when you are done. DefTool displays the new subkey group field under the group and its associated markers in the Definition Tree; note that the subkey group field icon looks different from the regular group field icon.
8. On the Group Properties dialog, define any additional group fields you want to index within the sub-account parent group. You can also define group fields in nested groups. You can define more than one group field for a group, but you must define only one as the subkey. You can use a group field in multiple locations.
9. Use the DefTool Simulator, described below, to test extraction of subkeys and group table data from your data input file. To edit data extraction rules, right-click a group or on a marker in the Definition Tree and select **Edit** from the right-click menu. To edit extraction rules for subkey or other group field, see "Modifying a group field or subkey " in this section.
10. Save the DDF.

**To simulate extraction of sub-account group data and subkeys in DefTool:**

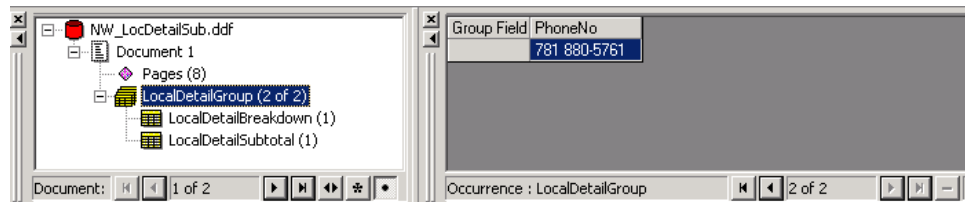
1. Click the Extractor Simulation icon  in the Simulation toolbar or select Extractor Simulation from the Simulator menu. The Simulation pane appears at the bottom of the DefTool main window. Nodes shown in the tree on the left represent each document detected in the input file. On the right, Simulator displays the document boundaries and extracted fields.



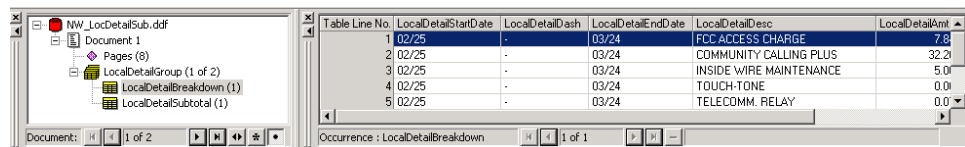
2. Click the document to expand the list of pages, groups, and fields.



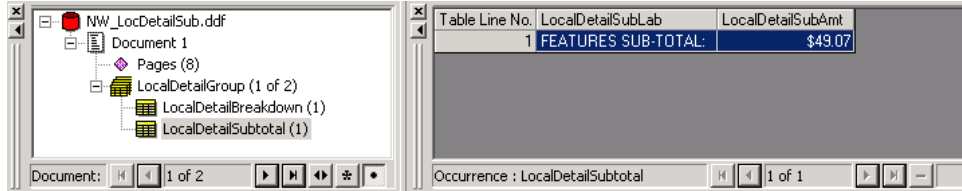
3. Click the sub-document group to display the subkey or group fields for the first occurrence.




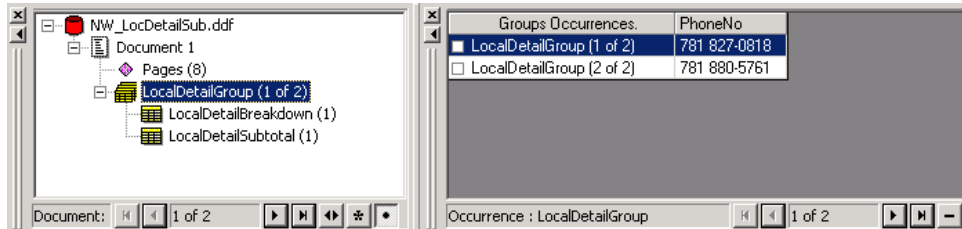
4. To display group data extraction, click the table.



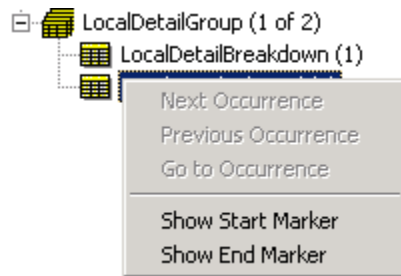
Sub-Account Indexing




- To test the extraction of all subkeys, click the **Show All Tables** icon , click the Document to expand the list of fields, tables, and groups, then click the sub-document data group. DefTool displays the extracted subkey and group fields for each group occurrence in the document.



- You can use all the existing features of the DefTool Simulator to test the groups, tables, and markers. For example, you can right-click on a group or table to display a menu of options:



- As with any groups, tables, or fields, you can choose to selectively simulate a sub-document group (using the Simulator Settings icon  or the Simulator Settings option from the Simulator menu, Group tab). By default, DefTool simulates all tables, group fields, and subkeys associated with the group.



## Modifying a Group Field Or Subkey Properties

You can edit the properties of a group field, including a subkey, after you have created it. For example, the simulation process may reveal a problem that requires you to edit the extraction rules for a group field, such as the field pattern or window coordinates.

### To edit the properties of a group field:

1. With the DDF open in DefTool, expand the Definition Tree to expose the group field.
2. Right-click on the group field (can be a subkey), and select **Edit** from the right-click menu. The Group Field Properties Dialog appears.
3. Make your edits using the Group Field Properties Dialog and click **OK**.

## Deleting a Group Field Or Subkey

You can delete a group field, including subkeys, from a DDF. If a group field or subkey has multiple locations, you can select and delete by location (similar to how fields work).

### To delete a group field or subkey:

1. With the DDF open in DefTool, expand the Definition Tree to expose the group field.
2. Right-click on the group field (can be a subkey), and select Delete from the right-click menu. DefTool asks if you are sure you want to delete the field.
3. Click **Yes**. DefTool deletes the group field.

### To delete multiple group fields:

1. With the DDF open in DefTool, expand the Definition Tree to expose the group.

2. Right-click the group and select **Edit** from the right-click menu. DefTool displays the Group Properties dialog.
3. Under the Included Group Fields, select a field to delete and click Delete. DefTool asks if you are sure you want to delete the field.
4. Click **Yes** to delete the field. DefTool deletes the group field.
5. Continue deleting group fields as necessary.
6. Click **OK**.

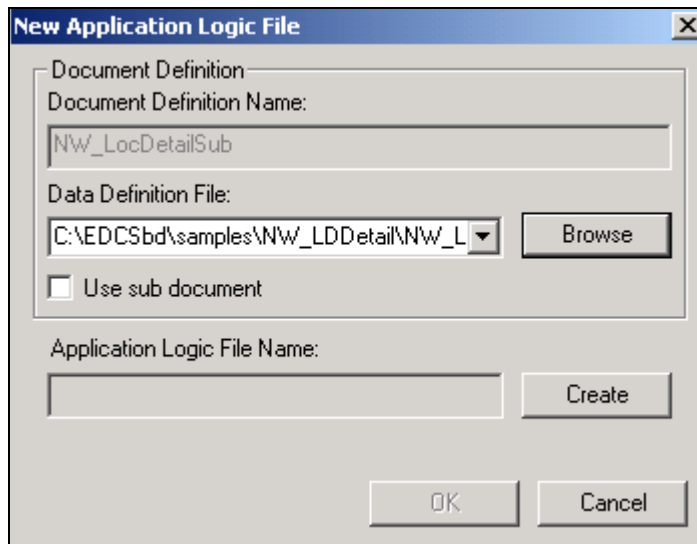
## Creating an ALF for the Sub-Account View in Composer

To use the sub-document indexing feature you must create and use a separate Application Logic File (ALF) for presenting the sub-document view. This is true for both new eaDirect applications and for adding the feature to an existing application.

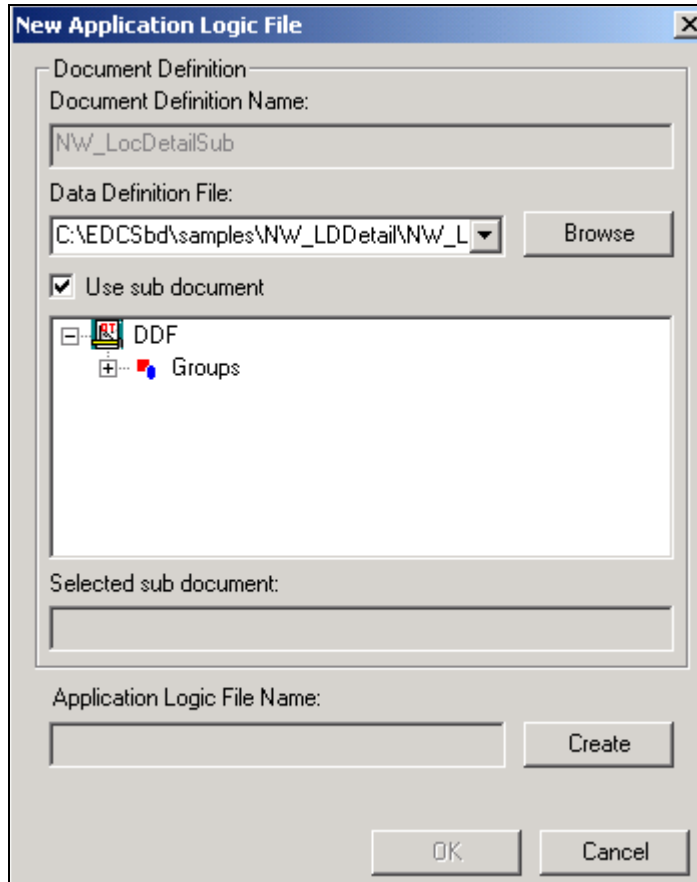
You must map the group and subkey to the default template. (You do not need to specify a group template for the sub-documents; the group becomes the main document in this view.)

**To create a new sub-account ALF and map the data for presentment:**


1. In Composer, select New ALF from the File menu. Composer displays the New Application Logic File dialog. Select the application DDF you created for this view.

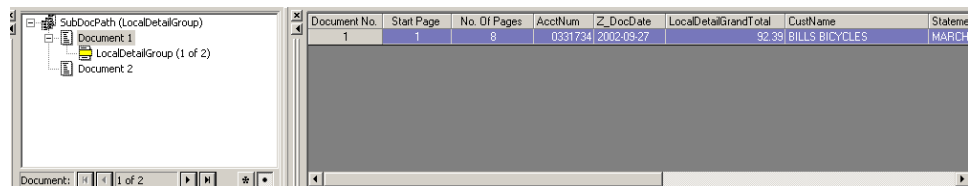


2. Select the **Use sub-document** option. Composer displays a tree showing the sub-document groups in the DDF.



3. Click on the **Groups** node to expand the list. Select the group containing the subkey for the sub-account you want to use in this ALF view.
4. Click **Create** and specify a name for the ALF, then click **Save**. Click **OK**. Note that in the Definition Tree, Composer converts the sub-document group into pure tables, fields into pure fields, and the subkey to the primary key. Composer treats the sub-document as a normal document, not a group (and there is no group template for the group).

5. In the Application Tree, right-click the Home node and select Add Default Template from the right-click menu. The New Default Template dialog appears. Select the template you to use as the default, and click **OK**. Composer opens the selected template.
6. Drag-and-drop the tables and groups to the template as you want them to appear in the view.
7. Click  to save the ALF.
8. You can use the Composer Simulator to verify that the composed document appears as you intend. Expand the document node in the left pane to display the sub-documents.



### To delete a group field or subkey from a template:

- With the ALF open in Composer (WYSIWYG view), select the group field or subkey E tag, for example **[E] PhoneNo, F [/E]**, and delete it. (This is how you currently delete tables and fields.)

## Applying Sorting, Filtering, and Other Features Within Sub-Accounts

You can apply the following features within sub-documents in much the same way you apply them to regular (parent) documents. You can apply:

- **Sorting, filtering, arithmetic subtotalling, combine, and paging elements** to tables or groups within a sub-document, but not on a group field.
- **Dynamic patterns** to pass patterns to any child group markers within the sub-document group, but not to a group field.

- **Links** to a group field, including a subkey.
- **Conditional logic** on tables and groups within the sub-document, but not on a group field.

## Purging Sub-Account Index Data

eaDirect purges sub-document index data from the database when purging the data for its parent, or root document.

Use the **Purge Apps** Command Center job to purge data. If you have sub-document data, the **PurgeIndexSummary** task of the Purge Apps job automatically purges the related sub-document data. See the *eaDirect Production Guide* for more information about purging data.

## APIs for Sub-Account Indexing

eaDirect 3.2 includes new signatures of APIs in the Content Access and Verify SDK packages to support sub-account indexing for custom content access and for custom audits of indexed volumes.

The new APIs use one of the following parameters (instead of a string):

Name [] `accounts`

Name `account`

**Name** represents a generic name, in this case the hierarchical sub-document account path; **account** or **accounts** represent the sub-account number.

See the Javadoc included with 3.2 eaDirect for details on using these APIs.

### `com.edocs.app.user`

The `com.edocs.app.user` API lets you customize content access and history lists in your eaDirect application.

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, subtotalling, and updating optional fields at presentment.

Use the new APIs listed below to modify your application to add access to sub-documents and include sub-account data in a history list. These new APIs use the new parameter shown for sub-documents:

<b>API</b>	<b>Subdocument Parameters</b>
SendDocument	Name[] accounts
GetSummary	Name account
UpdateSummaryInfo	Name account
getDocumentReader	Name[] accounts
GetDocumentInputStream	Name[] accounts

## com.edocs.app.verify

The com.edocs.app.verify API lets you implement the custom Audit to Verify feature in your eaDirect application. This feature let users audit indexed data volumes before releasing them for presentment.

This Verify class is the interface to eaDirect core for managing indexed files or volumes. Use the new Verify APIs listed below to modify your application to include sub-document indexing with this feature. These APIs use the new parameter shown for sub-documents:

```
getHitList
Name account
```





# Enterprise Profile Management



The eaSuite includes powerful new tools for customer self-care with eaCare. This chapter provides a summary of the eaSuite features that support enterprise profile management (EPM) for B2B hierarchies:

- The Hierarchy Console web application
- The APIs provided with the Hierarchy Console to customize the console

## Using The Hierarchy Console For Enterprise Profile Management

The Hierarchy Console provides an intuitive user interface for administrators to organize and edit account information. Its flexible format is open for customization and import/export aware through edocs Common Directory Access (CDA). Use the Hierarchy Console to map accounts and subaccounts into account groups that can control access privileges, statement viewing, editing user profiles, and other online account management.



The Hierarchy Console web application (EAR file) extends the eaDirect user management framework (UMF) to support a flexible B2B hierarchy. The Hierarchy web application and its APIs support:

- Large and complex account statements
- B2B ways of organizing account information (detail records, departments, cost centers, documents and subdocuments) not contained in the data source
- Changes to organization structures, cost centers and divisions that are charged back through internal A/R processes for billed services. These changes need not be present in the data source, for example, when changes result from acquisition, restructuring, or consolidation.

Direct billing to customers (B2C) usually involves simple relationships between statements and users. One customer has one account number where they log in to view one statement each month. Business-to-Business companies (B2B) generally have much more complex relationships with their customers.

B2B customers may have one or many users looking at one or many statements or pieces of statements. These users will also have different privileges for what content in the statements they can view, and what actions they may take on that content, for example to approve or pay it.

The **Hierarchy Console** lets you define and describe these users, privileges, and actions through a simple web console interface. For example, you can use the hierarchy Console to design and apply a **B2B hierarchy** of cost centers. The eaSuite can then organize, extract, and display a company phone bill by cost center, so that each manager could view, approve, and pay only their department's charges. This process of organizing and displaying subdocuments is sometimes called “bursting” an invoice or statement.

**Tip**

The Hierarchy Console does not perform any calculations directly. Rather, it passes data and metadata to other components that calculate statement data for presentment.

The Hierarchy Console itself does not control access to statements. Rather, an administrator can assign a user or account to a folder in the hierarchy, so that anyone with access privileges to that folder may use eaDirect to view statements for users in that folder.

## Setting Up the Hierarchy Console

This procedure assumes you have installed eaDirect and the web application archive `ear-hierarchySample.ear`.

### To set up the Hierarchy Console:

1. Create the CDA schema with the script `create_hierarchy_schema`. This creates the root node `o=Hierarchy`.
2. Deploy the sample EAR file `ear-hierarchySample.ear`.
3. Index the HierarchySample data by creating an eaDirect application and running the Indexer job on the sample data file.

4. Navigate to the Hierarchy Console at your application server and port, for example: `http://dusky:7001/hierarchySample/user/jsp/index.jsp`

**Caution**

The `o=Hierarchy` name is the *context root* of the hierarchySample web application. You cannot change this name without making a deployment descriptor change to `ejb-enrollment-hm.jar`. In `META-INF/ejb-jar.xml`, change the `contextRoot` value to your new context root. For more information on customizing application context, see the SDK **User Management Framework**.

5. Enroll an administrative user as a SuperAdmin with root level privileges.

For details on creating folders and enrolling users, see [Working with Hierarchies](#).

## Adding Attributes to the CDA Schema

In this step, you set your classpath and run the CDA client to create the CDA schema for the Hierarchy Console. This procedure assumes you have created an eaDirect database as part of your installation.

You need classpath, environment, and password information from your database installation for this procedure. Write down your information in this table, and contact your system administrator if you need help.

Variable	Default/Example Values	Your Values Here
Login user	oracle	
Login password	edocs	
EDX_HOME	/opt/EDCSbd/	
WL_HOME or WS_HOME	/opt/BEA/wlserver6.1	
JAVA_HOME	/export/home/bean/jdk131	
Server name	dusky	
Database port	1521	
Database instance	edx0	
DBA name	edx_dba	
DBA password	edx	

### To create the default CDA schema for the hierarchy Console:

1. Set your environment with the shell script `setEnv.sh`, for example:

```
. $WL_HOME//config/mydomain/setEnv.sh
```

2. Log in as the edocs user (**nobody**) and set your CLASSPATH to include your edocs, application server, and Java home directories. For example:

```
CLASSPATH=$EDX_HOME/lib/edx_common.jar:$WL_HOME/lib/weblogic.jar:$WL_HOME /jdk131
```

---


**Tip**

If your JDK is set elsewhere in your environment, you may not need to include it in your classpath. If you receive errors while running the script, set your classpath to include your JDK.

---

3. Export your CLASSPATH.
4. Navigate to the sample directory for `create_hierarchy_schema` at `<EDX_HOME>/samples/hierarchySample/schema`
5. To start the CDA client and create the default CDA schema for the hierarchy, edit this command for your own environment with the values for your database server name and port and oracle username, password, and database instance.

```
java -Djdbc.drivers=oracle.jdbc.driver.OracleDriver
com.edocs.jndi.cda.cli.Main
jdbc:oracle:thin:@dusky:1521:edx0 edx_dba edx <
create_hierarchy_schema
```

6. Run your edited command. If the script executes successfully, you will see a series of caret prompts, like this: `>>>>>>>>>>`
7. If your application server is running, stop and start the server to pick up the changes to the CDA schema.
8. If you run the `create_hierarchy_schema` script again, or if you run another database script such as `create_training_schema`, you may need to remove or comment out any schema binding conflicts. For example, both these scripts create an `sb role` attribute, and the second script will throw a `SchemaBind` exception.



---

The application server caches schema information. If you make changes while the application server is running, you must stop and start your application server to pick up the changes to the CDA schema. If you receive errors enrolling a SuperAdmin, stop and start the app server before enrolling users.

---

By default, the `create_hierarchy_schema` script pre-populates the Hierarchy Console with a root level organization “Hierarchy.” You then use the Hierarchy Console to create your own hierarchy subfolders under this root level. You can then map account and user information from subdocuments into your hierarchy to group and organize accounts.

You must first create a root-level hierarchy folder (or run `create_hierarchy_schema`) and enroll at least one SuperAdmin user with root level privileges before you can assign users and accounts to folders. For more information, see [Creating Hierarchy folders](#).

## Deploying the Hierarchy Web Application

Start your application server if it is not already running, and deploy the sample EAR file

```
<EDX_HOME>/samples/hierarchySample/J2EEApps/weblogic/ear-hierarchySample.ear
```

## Indexing Hierarchy Sample Data

Your eaDirect installation places a sample DDF, ALF, and dataset for the Hierarchy Console in

```
<EDX_HOME>/samples/NW_Subdoc
```

Use the files in this folder to create an eaDirect application and run the Indexer job.

The Indexer job indexes all **subdocuments** defined in your DDF. For more information, see **Subdocument Indexing**.

## Working with Hierarchies

### SuperAdmin and Admin Roles and Privileges

A **SuperAdmin** *registers* **primary accounts** with an **organization**. When a SuperAdmin logs in, the left pane of the Hierarchy Console presents a list of *unregistered* accounts containing all primary accounts for the DDN, for example IBM. These primary accounts can then be associated with an organization within the hierarchy, for example Northeast or Central.

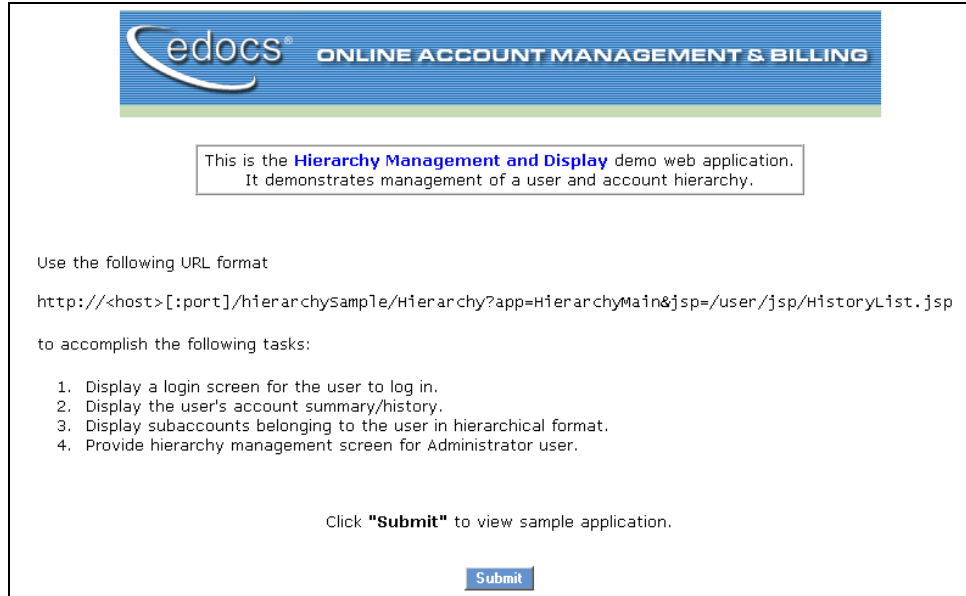
An Admin *assigns* **subaccounts** into one or more organizations within the hierarchy. When an Admin logs in, the left pane of the Hierarchy Console presents a list of *unassigned* subaccounts for the Admin's organization(s), from all primary accounts that the SuperAdmin has registered with that organization. For example, these subaccounts could be the telephone numbers of IBM employees in the Northeast, which the Admin could then assign to the Boston or New York organizations.

### Logging In to the Hierarchy Console

To bring up the Hierarchy login screen, use the following URL format for your own host and port:

```
http://<host>[:port]/hierarchySample/user/jsp/index.jsp
```

## Enterprise Profile Management



**edocs**® ONLINE ACCOUNT MANAGEMENT & BILLING

This is the **Hierarchy Management and Display** demo web application. It demonstrates management of a user and account hierarchy.

Use the following URL format

`http://<host>[:port]/hierarchySample/Hierarchy?app=HierarchyMain&jsp=/user/jsp/HistoryList.jsp`

to accomplish the following tasks:

1. Display a login screen for the user to log in.
2. Display the user's account summary/history.
3. Display subaccounts belonging to the user in hierarchical format.
4. Provide hierarchy management screen for Administrator user.

Click **"Submit"** to view sample application.

**Submit**

Clicking the Submit button brings you to the login screen.



**edocs**® ONLINE ACCOUNT MANAGEMENT & BILLING

**Enroll Now**

**User Login...** Enter your username and password, then click "Submit".

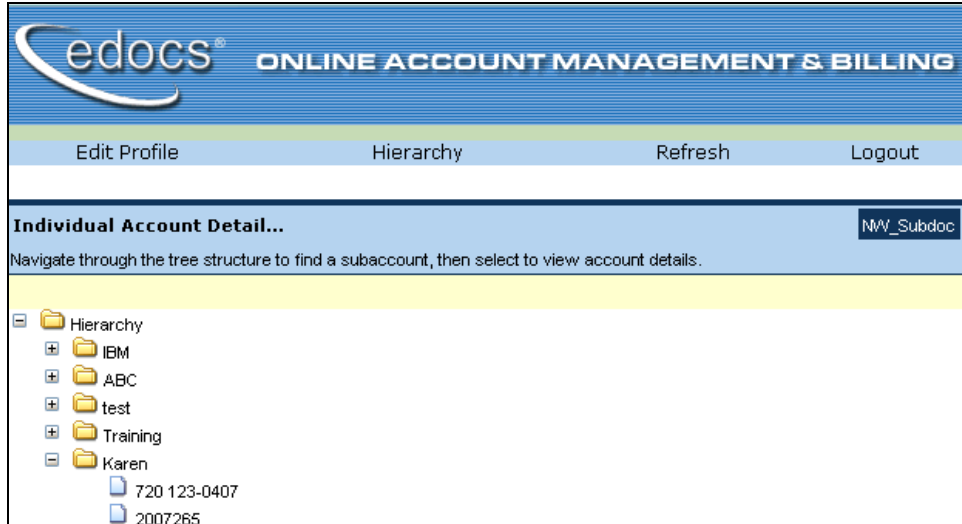
<b>Username:</b>	<input type="text" value="SU"/>
<b>Password:</b>	<input type="password" value="*****"/>
<b>DDN:</b>	<input type="text" value="NW_Subdoc"/>

© Copyright 1997-2002 edocs®, Inc. All Rights Reserved.  
edocs is Reg. U.S. Pat. & Tm. Off. [Privacy Policy](#)

From this screen, you can log in as an existing user or enroll a new user. For enrollment, see [Enrolling Users in the Hierarchy](#).

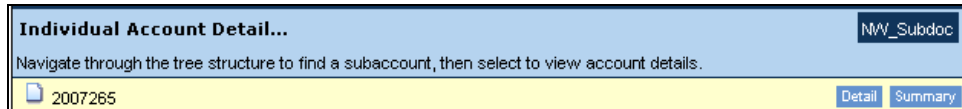
When a user logs in, she sees her assigned level in the hierarchy and may navigate down through any subtrees to view any accounts for which she has privileges.

For example, a **SuperAdmin** user has privileges to view all organizations in the Individual Account Detail pane, and may register primary accounts to these **organizations**. In this example, the SuperAdmin user **su** has logged in to the Hierarchy Console and can see the root level folder **Hierarchy** and all folders below it.



You must enroll at least one SuperAdmin user before enrolling users at lower levels in the hierarchy. For more information, see [Enrolling Users in the Hierarchy](#).

Select an account from the tree. It appears as a highlighted bar at the top of the Console pane.



Click the **Summary** button to view account data for this account.

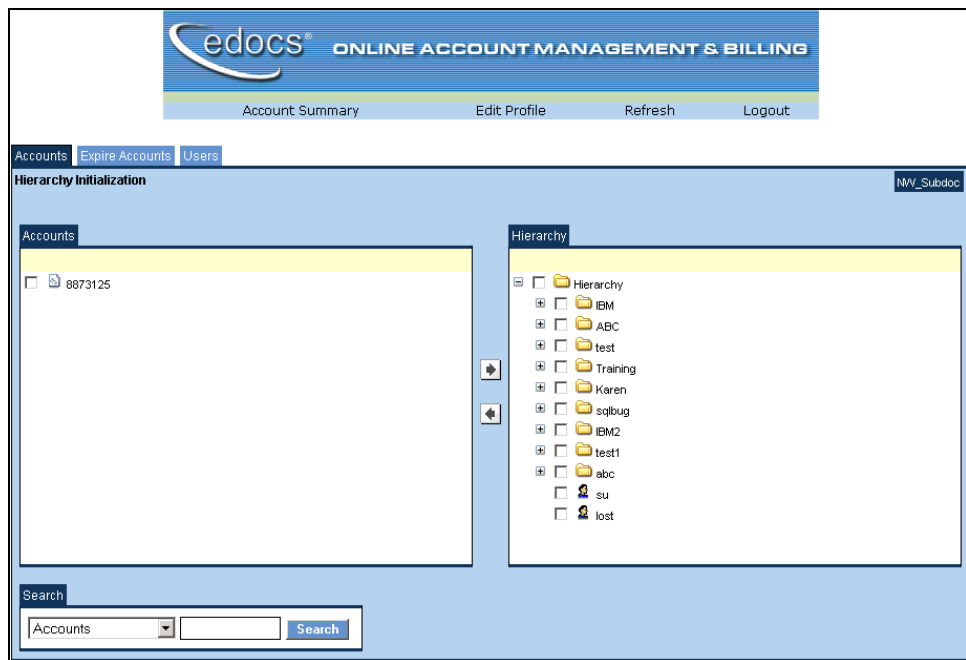
Account History for 2007265... Click the view icon to display account detail.							
View	DocumentDate	CustName	StatementDate	AmtDue	State	CustType	DueDate
	10-10-2002	BLACK & BROWN CUSTOM LOG CABIN	MARCH 25, 2001	186.67	NM	B2B	04/19/01 TO
	10-10-2002	BLACK & BROWN CUSTOM LOG CABIN		186.67	NM		04/19/01 TO
	09-27-2002	BLACK & BROWN CUSTOM LOG CABIN		186.67	NM		04/19/01 TO

From the login screen, click **Hierarchy** in the navigation bar to go to the Hierarchy Console.

## Navigating the Hierarchy Console

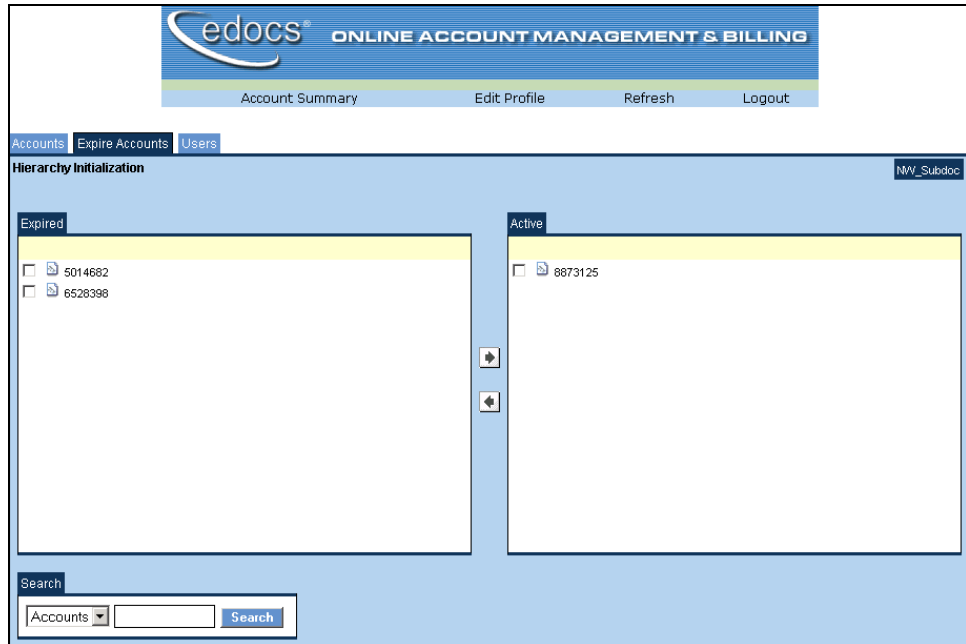
The Hierarchy Console has two panes, each containing a directory tree of nested folders, users, and accounts. The left pane has three tabs: **Accounts**, **Expire Accounts**, and **Users**.

Click the **Accounts** tab to show all accounts not assigned to a folder in the hierarchy. To assign accounts, see [Assigning accounts into hierarchy folders](#).

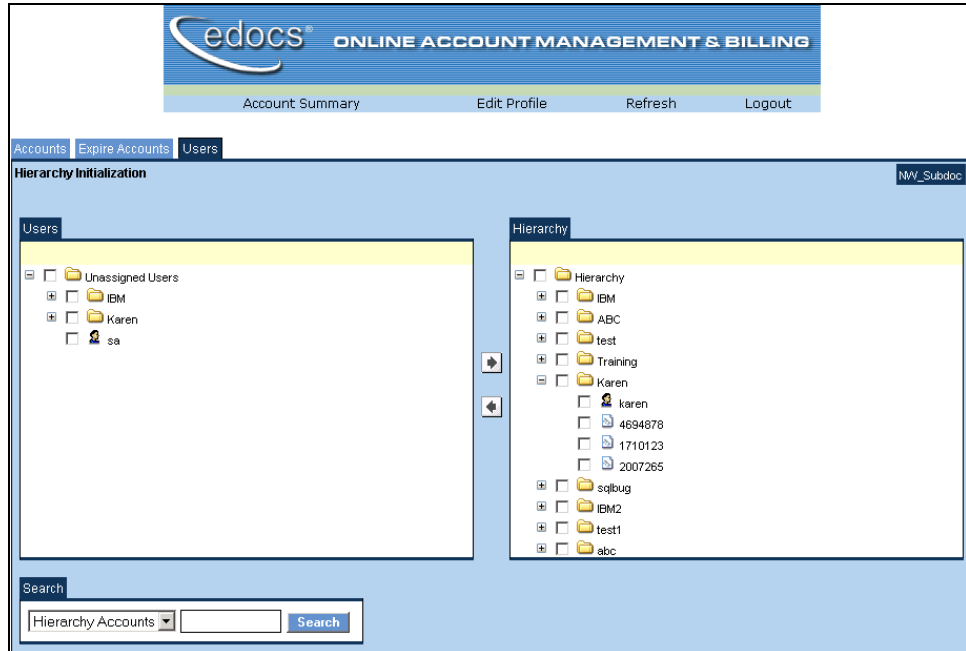


Click the **Expire Accounts** tab to show a list of expired and active accounts. To change the status of an account, select its checkbox and use the left or right arrow.

## Enterprise Profile Management



Click the **Users** tab to show all users (including any Admins and SuperAdmins) not assigned to a folder in the hierarchy. To assign users, see [Assigning users into hierarchy folders](#).



In the right pane, the **Hierarchy** tree presents a directory of **nodes** in the hierarchy as folders. Clicking a folder expands it to show all accounts and users assigned to that node in the hierarchy.

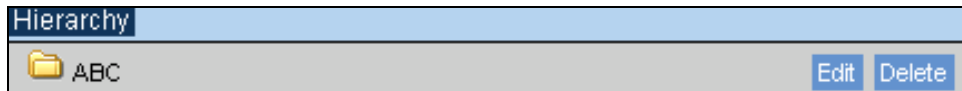
## Creating Hierarchy Folders

An Admin or SuperAdmin may create folders at any level in the hierarchy for which they have privileges. A SuperAdmin must create at least a top-level folder for each organizational hierarchy before anyone can assign users or accounts.

### To create a folder in the Hierarchy Console:

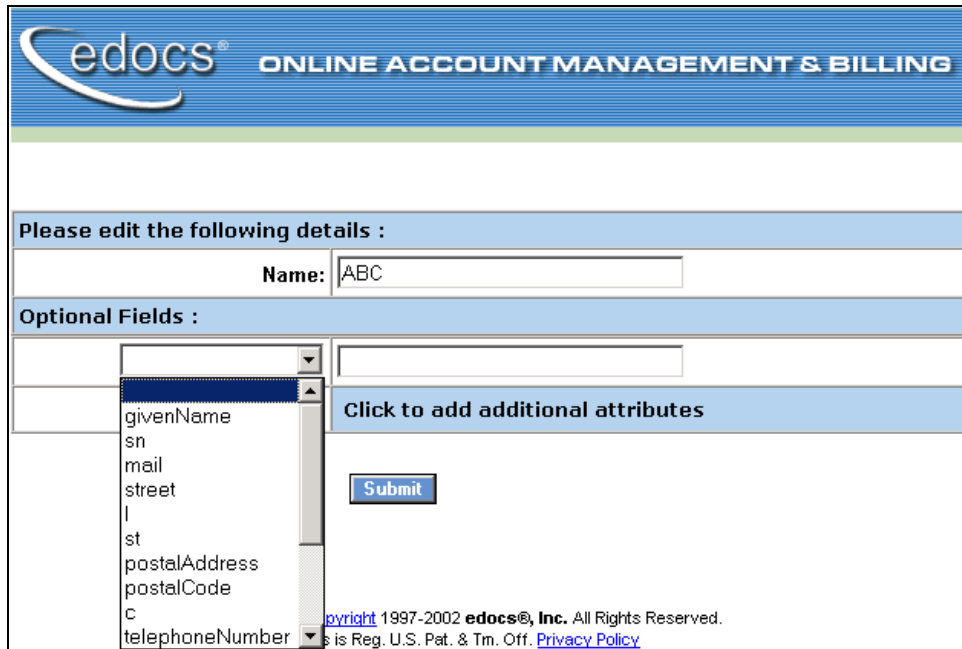
1. Bring up the hierarchy console and log in with your username, password, and DDN to go to the login page.

2. You will see the context root folder **Hierarchy**, created by the script `create_hierarchy_schema`. Within this root folder, you may then create subfolders for your Organizational Units, (ex. ABC, Gateway).
3. Clicking a folder highlights it at the top of the page with Edit and Delete buttons.



## Editing Hierarchy Folders

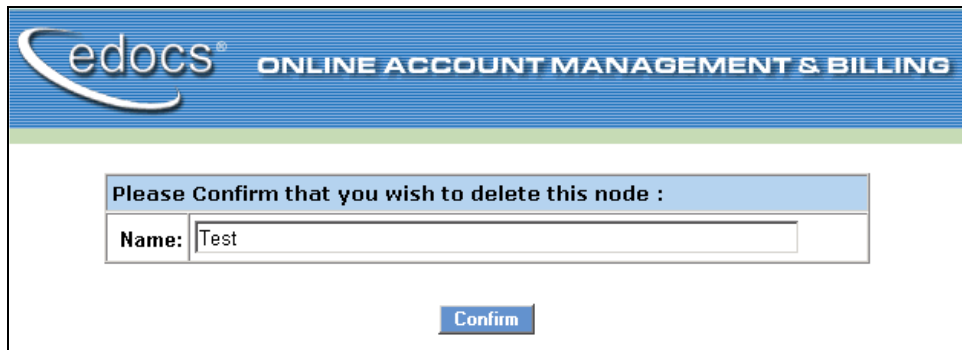
Highlight the folder and click **Edit**. You can add additional attributes to a folder through this screen.

A screenshot of the edocs ONLINE ACCOUNT MANAGEMENT & BILLING interface. The page has a blue header with the edocs logo and the text "ONLINE ACCOUNT MANAGEMENT & BILLING". Below the header, there is a section titled "Please edit the following details :". Under this section, there is a "Name:" label followed by a text input field containing "ABC". Below this is a section titled "Optional Fields :". This section contains a dropdown menu with a list of attributes: givenName, sn, mail, street, l, st, postalAddress, postalCode, C, and telephoneNumber. To the right of the dropdown menu is a text input field and a button labeled "Click to add additional attributes". Below these elements is a "Submit" button. At the bottom of the page, there is a copyright notice: "Copyright 1997-2002 edocs®, Inc. All Rights Reserved. This is Reg. U.S. Pat. & Tm. Off. Privacy Policy".

## Deleting Hierarchy Folders

An Admin or SuperAdmin may delete folders at any level in the hierarchy for which they have privileges. You may not delete a folder containing accounts or subfolders (children or leaf nodes). Before you can delete a folder, you must unassign all accounts and users and delete any subfolders.

Highlight the folder and click **Delete**. A confirmation screen appears. Click **Confirm** to delete the folder.



The screenshot shows a confirmation dialog box within the edocs interface. The header of the interface reads "edocs® ONLINE ACCOUNT MANAGEMENT & BILLING". The dialog box contains the following text and elements:

- Header: Please Confirm that you wish to delete this node :
- Label: Name:
- Text input field: Test
- Button: Confirm

## Enrolling Users in the Hierarchy

Before you enroll users, you should create folders for any Organizations to which you plan to assign your users. You will use the enrollment screen to create a new Organization.

To enroll a new user from the login screen, click **Enroll Now**. The enrollment page appears.

The screenshot shows the 'edocs ONLINE ACCOUNT MANAGEMENT & BILLING' interface. A central text box states: 'This **sample** demonstrates the flexibility of the **User Management Framework JNDI Implementation** by maintaining a hierarchical user enrollment schema.' Below this, a paragraph explains that users must select an organization from a list of active organizations. A link labeled 'here' is provided to create a new organization. The main form, titled 'Enter your subscription details:', contains the following fields: Username (text input), Password (text input), Re-Type Password (text input), Description (text input), User Role (dropdown menu with 'User' selected), Organization (dropdown menu with 'NatlWireless' selected), Email Address (text input), and Assign User into Hierarchy (dropdown menu with 'No' selected). At the bottom of the form are 'Submit' and 'Reset' buttons.

This table shows the allowable values for each enrollment field.

Field Name	Allowable Values	Status	Notes
Username	User-defined	Required	Enter the username you wish to create for this user. This is the primary account key.
Password	User-defined	Required	Enter the password you wish to create for this user. Retype the password for security.



Field Name	Allowable Values	Status	Notes
Description	User-defined	Optional	Enter an optional description of this user, for example their title or name.
User Role	SuperAdmin Admin User <i>(Names are for this sample and may be customized in a custom schema)</i>	Required	Select the level of the hierarchy you want this user to see. A SuperAdmin can see all folders from the root level. An Admin can see all folders at and below their assigned folder. A User can see only their own account information.
Organization	Hierarchy North South IBM Coke	Required	Select your DDN or use the Click Here link to create a new organization.
Email Address	User-defined	Optional	Enter the e-mail address for this user.
Assign User into Hierarchy	Yes/No	Required	Yes automatically adds this user to the organization (folder) you define. This allows this user to see the Hierarchy tab in the Console at the folder level you define on enrollment. No leaves this user unassigned. Later, a SuperAdmin or Admin for this hierarchy can assign unassociated users to a folder.

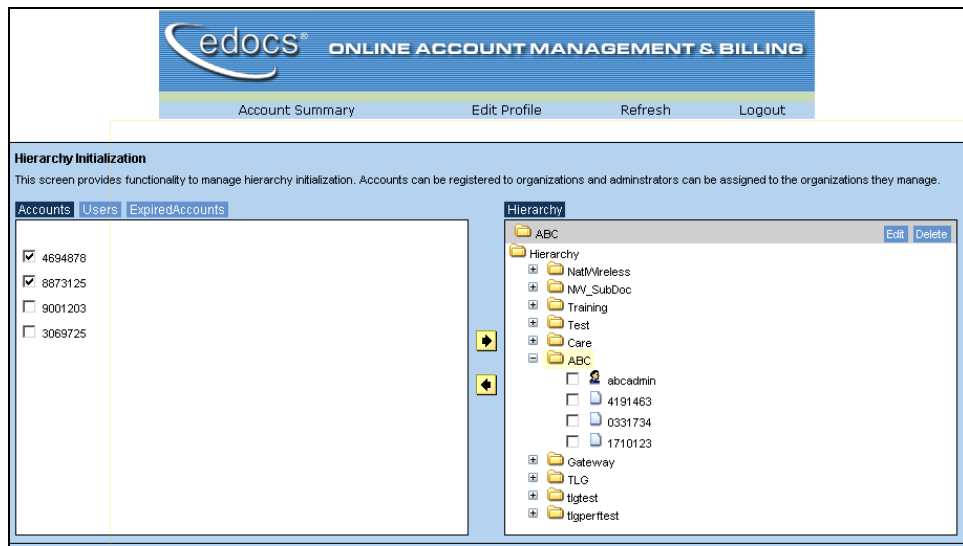
## Assigning Accounts Into Hierarchy Folders

Admins and SuperAdmins may assign some or all accounts to a folder at any level of the hierarchy *except the context root*. A folder may contain multiple accounts, but an account may be assigned to only one folder.

Admins and SuperAdmins may also remove an account from a folder. This does not remove the account from the system, but simply means the account is no longer associated with that folder. The dissociated account reappears in the list of unassigned accounts.

**To assign an account to a folder in the hierarchy:**

1. Log in as the SuperAdmin or Admin and click the Hierarchy link. The Hierarchy Console displays all folders for which you have privileges.
2. Click the **Accounts** tab in the left pane to see a list of unassigned accounts.



3. In the left pane, select the checkbox for the account(s) you want to assign.
4. In the right pane, select the folder to which you are assigning them. The folder will highlight as shown.
5. Click the **right arrow**. The account(s) will appear within the assigned folder.

## Assigning Users Into Hierarchy Folders

Admins and SuperAdmins may assign users to a folder at any level of the hierarchy, or assign a user directly to a single account. A folder may contain multiple users, but each user may be assigned to only one folder.

Admins and SuperAdmins may also remove a user from a folder. This does not remove the user from the system, but simply means the user is no longer associated with that folder.

### To assign a user to a folder in the hierarchy:

1. Log in as the SuperAdmin or Admin and click the Hierarchy link to go to the Hierarchy Console. You will see all folders for which you have privileges.
2. Click the **Users** tab in the left pane to see a list of unassigned users.

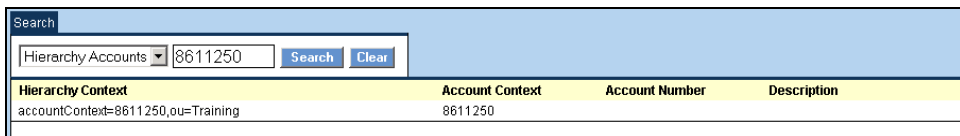
The screenshot shows the 'edocs ONLINE ACCOUNT MANAGEMENT & BILLING' interface. At the top, there are navigation links: Account Summary, Edit Profile, Refresh, and Logout. The main content area is titled 'Hierarchy Initialization' and includes a description: 'This screen provides functionality to manage hierarchy initialization. Accounts can be registered to organizations and administrators can be assigned to the organizations they manage.' Below this, there are three tabs: Accounts, Users, and ExpiredAccounts. The 'Users' tab is active, showing a list of 'All Unassigned Users' with columns for user name and organization. The 'Hierarchy' pane on the right shows a tree structure of folders, including 'ABC', 'NatlWireless', 'NW\_SubDoc', 'Training', 'Test', 'Care', and 'ABC'. A search bar is located at the bottom of the interface.

3. In the left pane, select the checkbox for the user(s) you want to assign.

4. In the right pane, select the folder to which you are assigning them. The folder will highlight as shown.
5. Click the **right arrow**. The user(s) will appear within the assigned folder.

## Searching for Accounts in the Hierarchy

To find an account in the hierarchy, use the Search box at the bottom of the Hierarchy Console screen. Select **Hierarchy Accounts** from the drop-down menu and enter an account number, then click **Search**. The Console displays the name of the Organizational Unit (ou) folder to which the account is assigned, in this example **Training**.



Hierarchy Context	Account Context	Account Number	Description
accountContext=8611250,ou=Training	8611250		

## Viewing Accounts For a User in the Hierarchy

### To view a list of accounts:

1. Enroll a User with the Hierarchy sample enrollment page (not the eaSample page).
2. Log in as that User and click the Account Summary link to go to the Individual Account Detail page.
3. The Individual Account Detail will display a list of available accounts, but not the Hierarchy assignment panes.

## Customizing the Hierarchy Console

The Hierarchy Console **must be customized** with the `com.edocs.hierarchy` API to support advanced features, including:

- Paying statements directly through the Hierarchy Console (viewing is controlled by each web application, paying requires a payment solution)
- Customizing the Hierarchy Console user interface
- Searching the hierarchy
- Support for multiple DDNs, for example to group the accounts of a single customer with ten accounts across three DDNs.

For custom solutions integrating the Hierarchy Console, please consult your edocs Professional Services representative.

## Application Programming Interfaces (APIs)

### `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 servlets in `com.edocs.app` for user login and content access to account data mapped to the hierarchy.

### `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.

For more information on Hierarchy APIs, please see the eaDirect Application Programming Interface Specification (Javadoc).

For example implementations of these APIs in the Hierarchy Console web application, please see the sample JSP pages in **ear-hierarchySample.ear**.

# Cutting, Copying, and Pasting DDF Entities

# 8

## About Cutting, Copying, and Pasting in a DDF

You can now cut, copy, and paste to move DDF entities within a DDF or to another DDF:

- Page styles
- Fields, including group fields
- Markers
- Tables and columns
- Groups

Using cut, copy, and paste saves time and reduces errors when building multiple application views.

(Note that you can only open a second DDF with a second data input file.)

If you copy and paste an entity into the same location as the original, you may be prompted to give it a new name. A "location" just refers to another place within the same DDF.

When you cut and paste an item, DefTool also moves or copies any associated child items (entities that belong to them) as child items in the tree.

This chapter describes basic cut/copy/paste rules for each type of DDF entity.

## Page Styles

When you copy and paste a page style (in the same DDF), DefTool prompts you to enter a new name. DefTool copies all the entities belonging to that page style onto the new one. Therefore, if a marker has a location in one page style, DefTool creates another marker in the target page style as well. (DefTool displays the marker under the root of the tree, not the node of the page style.)

## Fields, Group Fields

You can cut, copy, and paste fields within the same page style or to a different one. You can also cut, copy, and paste group fields from one group to another.

If you paste a field within the same page style, you must provide a new name for the field.

If you paste a field to another page style, DefTool adds it as another location, so you do not need to rename the field.

## Markers

You can move or copy a marker pair (start and end markers) to the same or another page style. DefTool moves or copies the associated tables and groups with the marker pair.

If you copy a marker pair to another page style, DefTool adds an additional location to the markers and displays the marker in the Definition Tree root (and not under the page styles).

If the source and target page styles are the same, DefTool adds a new location to each marker, and markers remain under their page node.

You can also move or copy individual start and end markers to the same or to another start or end marker.

Renaming a marker location is equivalent to renaming the marker.



## Tables, Columns

To copy a table, you must paste it to the marker pair you want to associate it with.

When you copy and paste a table, DefTool prompts you for a new name and uses this name as a prefix to the columns of the new table. DefTool displays the renamed table under the marker node in the Definition Tree.

When you paste a table, DefTool copies or moves the associated anchor and columns.

You can also cut or copy a column from one table to another table. DefTool prompts you for a new name. If you cut a column that is an internal anchor, the anchor becomes external.

## Groups

You can copy and paste a group to the same set of markers or to a new pair you want to associate the group with. When you copy and paste a group, you must provide a new group name (only the top-most group is a copy; the table and column names remain the same).

When you paste a group, DefTool moves all child groups and tables belonging to the group.

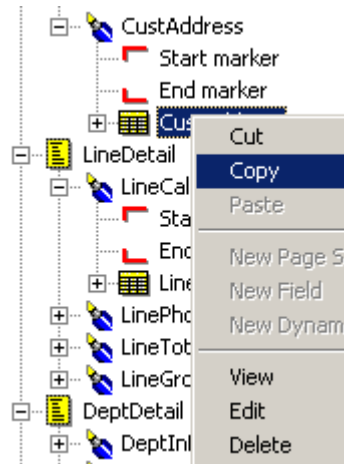
You can also copy a group and paste it as a child group under another parent group (in this case the group name stays the same).

## How to Move or Copy DDF Entities

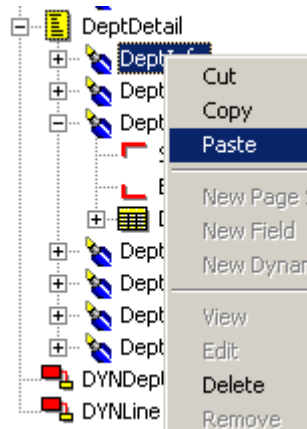
### To move or copy a DDF entity:

1. With the DDF open in DefTool, right-click the item in the tree you want to cut or copy and paste.

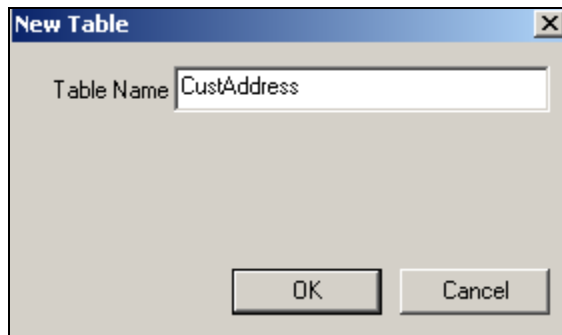
2. Choose **Cut** or **Copy** from the right-click menu.



3. Right-click the target location or item in the open DDF (or switch to another open DDF and right-click the target item there) and select **Paste** from the right-click menu.



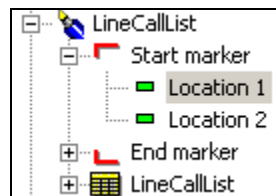
4. If DefTool prompts you for a new name, enter a unique name for the item and click **OK**.




5. DefTool redisplay the Definition Tree to reflect the result of your edits.



6. Note that if you copy and paste a marker pair to a different page style, DefTool displays the markers under the root, with multiple location nodes beneath it. To see which page style a location node represents, right-click on the node, such as "Location 1" and select **Edit** from the right-click menu. The **Marker Properties** dialog shows the page style name.



7. Click  to save the DDF.



## About Custom Post-Conversions

eaDirect 3.2 lets you create custom post-conversions to use with the data extraction rules you define using DefTool. When creating a new eaDirect application in DefTool, you define rules for extracting each element from your data input file for online presentment. The rules tell eaDirect where to find the data in the file (general location), the type of information it contains (content), and how to format the extracted data.

DefTool lets you apply post-conversion formatting specifications to a field or column value to further format the data for dynamic display online or for data storage purposes using the Command Center Indexer job. You can currently apply standard positional-based alphanumeric and date post-extraction conversions to extracted data, such as "Delete all trailing zeros or spaces(tm)" and "Translate Aug 21, 2002 to 08/21/2002(tm)." eaDirect 3.2 lets you create customized post-conversions to apply along with any sequence of existing (standard) post-conversions. You can apply multiple standard or custom post-conversions to a data field.

You can create custom post-conversions using:

- Regular expressions using the third-party PCRE 3.9 RegEx library for non-date and non-time data
- Date and time string formats
- New regular expression library

When creating a regular expression **for custom post-conversions** in eaDirect, you must use the syntax of the third-party PCRE 3.9 RegEx library. For information about this library and the use of regular expressions, see "Mastering Regular Expressions" by Jeffrey E.F. Friedl, available at <http://www.oreilly.com/catalog/regex/toc.html>



---

**Caution** The PCRE 3.9 RegEx library is not the same library you use to create regular expressions for extracting data using the DDF.

---

## Date and Time Formats

See "Appendix A - Data Definitions" in the *eaDirect User's Guide* for a list of date and time formats you can use to build post-conversions for date and time data in eaDirect.

## Reusing Custom Post-Conversions

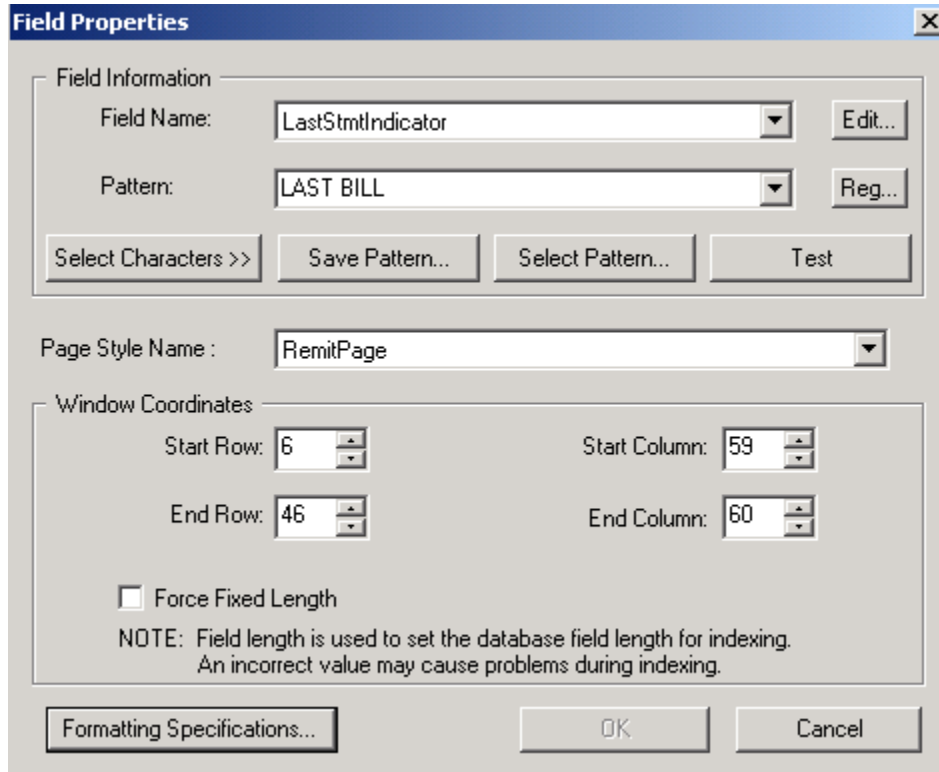
Each custom post-conversion you create becomes available for reuse with other data fields in the same eaDirect application (in the same DDF).

In addition, DefTool lets you save an application's custom post-conversions to an XML file. This enables you to individually copy and paste the expressions into custom post-conversions you create in other another DDF.

# Creating Custom Post-Conversions

**To create a custom post-conversion for a field or column:**

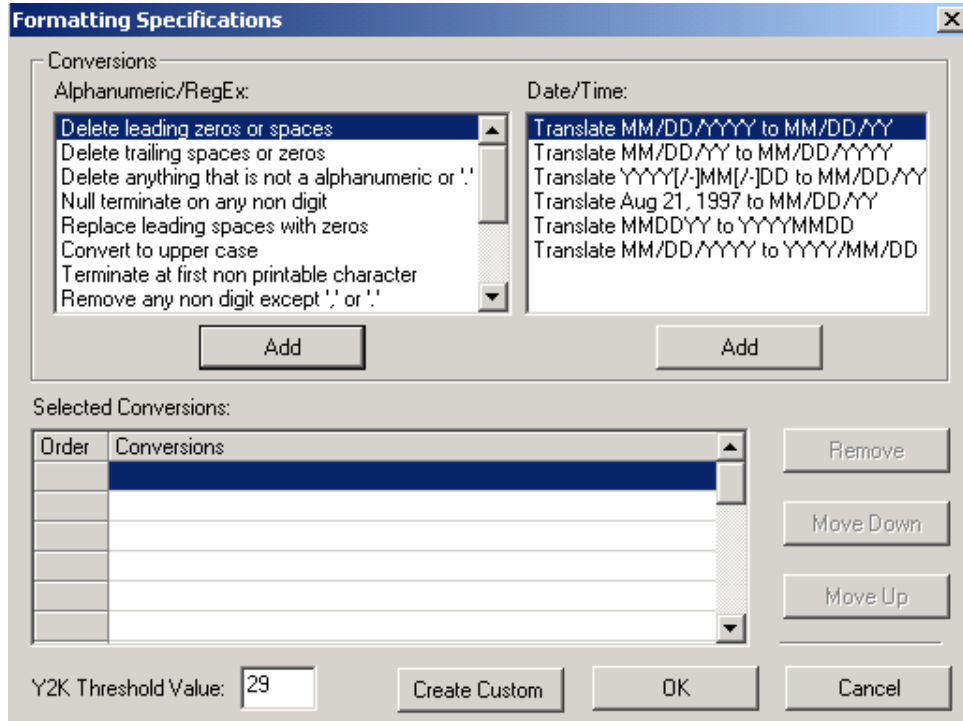
1. When defining the extraction rules for a field, DefTool displays the Field Properties dialog. (When defining a column, DefTool uses the similar Column Properties dialog.)



The image shows a 'Field Properties' dialog box with the following sections and controls:

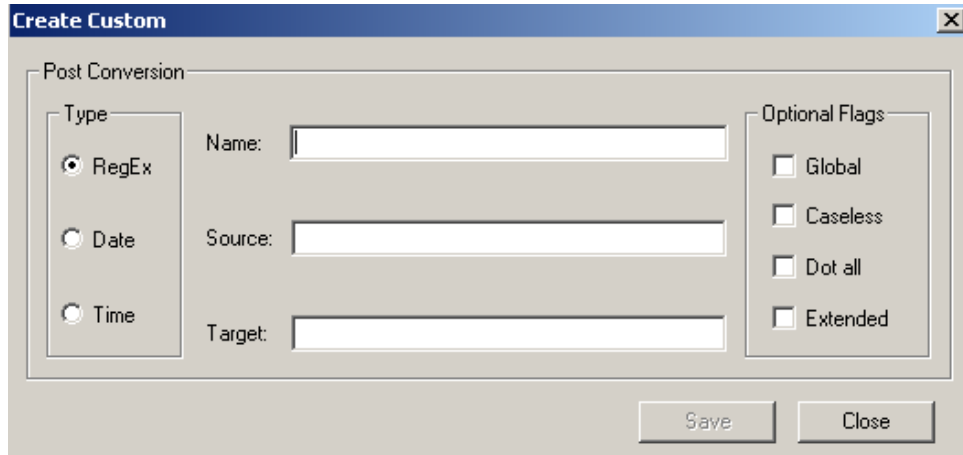
- Field Information:**
  - Field Name: LastStmIndicator (dropdown menu) with an 'Edit...' button.
  - Pattern: LAST BILL (dropdown menu) with a 'Reg...' button.
  - Buttons: 'Select Characters >>', 'Save Pattern...', 'Select Pattern...', and 'Test'.
- Page Style Name:** RemitPage (dropdown menu).
- Window Coordinates:**
  - Start Row: 6 (spin box)
  - Start Column: 59 (spin box)
  - End Row: 46 (spin box)
  - End Column: 60 (spin box)
- Force Fixed Length
- NOTE:** Field length is used to set the database field length for indexing. An incorrect value may cause problems during indexing.
- Buttons: 'Formatting Specifications...', 'OK', and 'Cancel'.

2. Click **Formatting Specifications**. DefTool displays the **Formatting Specifications** dialog. Use this dialog to choose any combination and sequence of standard or custom post-conversions to apply to the field or column.

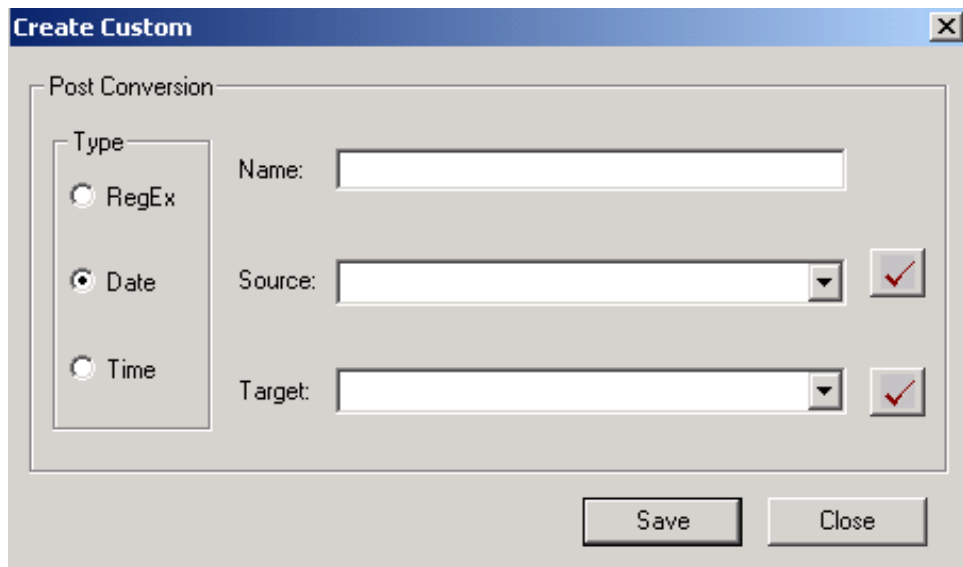




3. Click **Create Custom**. DefTool displays the **Create Custom** dialog.



4. Specify the type of post-conversion you want to create: **Date** or **Time** to convert most date or time data, or **RegEx** to specify a regular expression for all other data types. If you choose a date or time, the **Create Custom** dialog redisplay with different options.



## Custom Post-Conversions

5. Specify a meaningful name to identify the custom post-conversion. (This name appears in the "Alphanumeric/RegEx" or "Date/Time" conversion list on the Formatting Specifications dialog after you save, making it available to select with other fields or columns.)
6. In the Source field, enter the regular expression or date/time format functions you want to apply to the extracted data (this is on top of the regular expression defined on the Field Properties dialog that eaDirect first applies to the extracted data). For a regular expression, you can choose any of the following matching option letters; DefTool adds them to the expression between "(?" and ")".
  - g - Global
  - i - Caseless
  - s - Dotall
  - x - Extended
7. In the Target field, enter the expression for converting the extracted source data for presentment.
8. When creating a date or time post-conversion, click  after specifying the source and target functions to verify that the format is valid.

The screenshot shows a dialog box titled "Create Custom" with a close button (X) in the top right corner. The dialog is divided into a "Post Conversion" section and a "Type" section. The "Type" section has three radio buttons: "RegEx", "Date", and "Time". The "Date" radio button is selected. To the right of the "Type" section are three input fields: "Name:" (a text box), "Source:" (a dropdown menu with a checkmark button to its right), and "Target:" (a dropdown menu with a checkmark button to its right). At the bottom of the dialog are two buttons: "Save" and "Close".

9. When you are through specifying the post-conversion functions, click **Save**. Continue defining any additional post-conversions you need, then click **Close**. DefTool displays the Formatting Specifications dialog with the new custom post-conversion in the appropriate conversion list.
10. To add the custom post-conversion to the field, highlight the conversion and click Add. Click **OK** when you are done defining formatting specifications for this field.
11. You can continue the process of defining extraction rules for this field. See the *eaDirect User's Guide* for complete instructions.

## Editing a Post-Conversion

### To edit a custom post-conversion:

1. With the DDF open in DefTool, from the View menu, select **Custom List**. The **Custom Post Conversion List** dialog appears.
2. Highlight the custom post-conversion you want and click **Edit**. The **Edit Custom** dialog appears.
3. Enter your edits. Click **Save**.
4. Click **Close**.

## Removing a Custom Post-Conversion From a Field Or Column

### To delete a post-conversion (custom or standard):

1. With the DDF open in DefTool, right-click the field or column in the tree.
2. Select **Edit** from the right-click menu.
3. Click **Formatting Specifications**.
4. Highlight the custom post-conversion in the Selected Conversions list and click **Remove**.

5. Click **OK**.
6. Click **OK** on the Field (or Column) Properties dialog.

**To delete a custom post-conversion from the application:**

1. With the DDF open in DefTool, from the View menu, select **Custom List**. The Custom Post-Conversion List dialog appears.
2. Highlight the custom post-conversion you want and click **Delete**.
3. If the selected custom post-conversion is not in use, DefTool asks if you are sure you want to delete it. If it is in use, DefTool tells you which fields currently use the custom post-conversion and warns you that deleting it also removes it from the field formats. Click **Yes** to delete the custom post-conversion (or **No** to cancel). DefTool removes the custom post-conversion from the application and from any fields where it was in use.
4. Click **Close**.

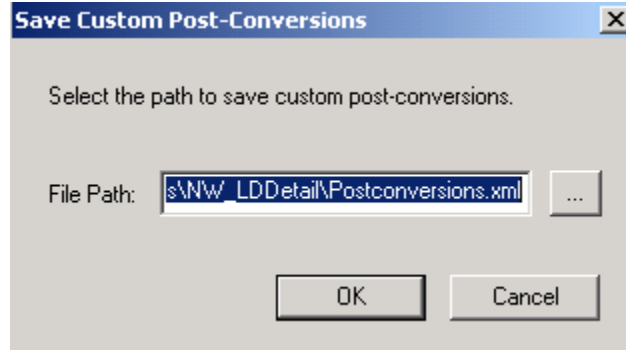
## Sharing Custom Post-Conversions With Other Applications

When you create and save a custom post-conversion, it becomes available for use in that application DDF only. To make those custom post-conversions available for use in another DDF, DefTool lets you save the custom post-conversions to an XML file. You must then recreate the custom post-conversions in the other DDF, copying and pasting the individual expressions from the file.

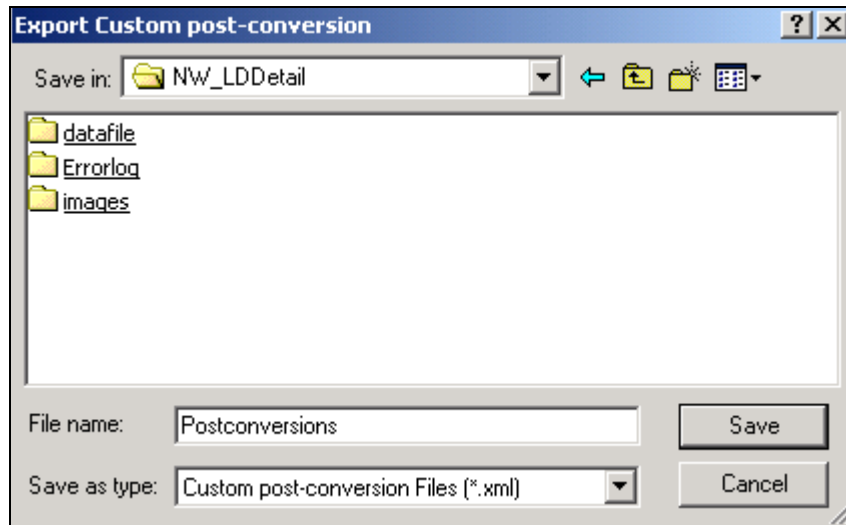
You can also add custom post-conversions to an existing file.

**To save custom post-conversions to an external text file:**

1. From the File menu, select **Save custom post-conversions**. The Save Custom Post-Conversions dialog appears.



2. Click  to browse for a directory and/or edit the file name. DefTool displays the **Export Custom Post-Conversion** dialog.



3. Specify the path and/or file name and click **Save**. Click **OK**.

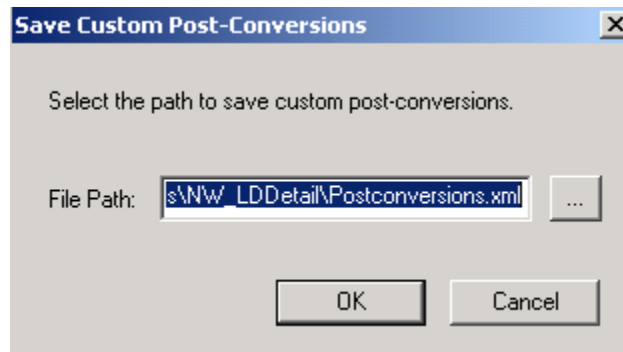
**To copy saved custom post-conversions to another DDF:**


1. In DefTool, open the DDF to which you want to add the saved custom post-conversions.

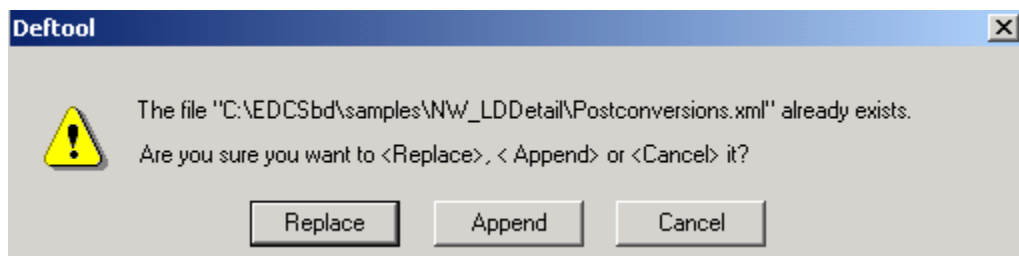
2. In a text editor, open the XML file containing the saved custom post-conversions.
3. Following the instructions in this chapter to create custom post-conversions, copy and paste the individual expressions from the XML file to recreate the custom post-conversions in this DDF.

**To append or replace an existing custom-post-conversions file:**

1. From the File menu, select **Save custom post-conversions**. The Save Custom Post-Conversions dialog appears.



2. Click  to browse for a directory and/or edit the file name. DefTool displays a warning dialog.



3. Click **Append** or **Replace**. **Append** will add any new custom post-conversions to the DDF, while **Replace** replaces the file contents entirely.

## Date and Time Format Constraints

See "Appendix A: Data Definitions" in the *eaDirect User's Guide* for details about the basic date and time formats available.

You must observe the following restrictions when using date and time formats to create custom post-conversions in DefTool:

- When converting a time duration format to another duration format, use only the following string components: %s, %Q, %K and %G
- You cannot convert a time format to a time duration or vice-versa. For example, you cannot convert the time duration "%K:%Q" to time format "%H:%M:%S" or the reverse. (It is OK to convert one time format to another time format, or to convert one time duration to another time duration, however.)
- Do not include both time duration formats and time formats in either the source or target. For example, you cannot have a source format of "%H:%Q:%S" and make the target "%T" because "%Q" represents duration and "%H" is a time format.
- You cannot convert minutes to hours. Since the clock is in 1000 minutes, you can specify only %Q for the source. This only stores minutes in the tm structure and will contain values up to 999 and does not populate the hour field.

## Examples of Custom Post-Conversions

The following examples show the regular expressions and formats you would use in the Source and Target fields on the Create Custom dialog to create various custom post-conversions.

Post-Conversion	Description	Source (RegEx)	Target (Format)	Flag (Mode)
1,234,567.89 to 1.234.567,89	Convert dot(.) to comma(,) and comma(,) to dot(.)	(\,){1} (\.)	(?1)(?2.)	/g (Global)

Custom Post-Conversions

Post-Conversion	Description	Source (RegEx)	Target (Format)	Flag (Mode)
1234567.89 to 1.234.567,89	Insert thousand separator as dot(.) and decimal separator as comma(,)	(\.)\{1\}\ G\{d\{	(?1,)(?2\$&.)	/g (Global)
00001234567.89 to 1.234.567,89	Remove leading zeros; insert thousand separator as dot(.); change decimal separator to comma(,)	(\.)\{1\}\ ^0+\ G\{d\{1,3\}\} (?=(?:\d\d\d)+(?!\d))	(?1,)(?2)(?3\$&.)	/g (Global)
-1234567.89 to -1.234.567,89	Maintain negative sign; insert thousand separator as dot(.); change decimal separator to comma(,)	(\.)\{1\}\ G\{-? (\d\{1,3\}\}(?=(?:\d\d\d)+ (?!d))	(?1,)(?2\$&.)	/g (Global)
-1.234.567,89	Maintain negative sign; change thousand separator to dot(.) and decimal separator to comma(,)	(\.)\{1\}\ (,)	(?1,)(?2.)	/g (Global)
€1,234,567.89 to €1.234.567,89	Maintain the currency sign; change decimal separator to comma(,) and thousand separator to dot(.)	(\.)\{1\}\ (,)	(?1,)(?2.)	/g (Global)
€-78912354354.65767 to €-78.912.354.354,65767	Maintain currency sign; change decimal separator to comma(,); insert thousand separator as dot(.)	(\.)\{1\}\ G\{€?\ -\ ? (\d\{1,3\}\}(?=(?:\d\d\d )+(?!d))	(?1,)(?2\$&.)	/g (Global)
€-000058736545656.74647467 to €-58.736.545.656,74647467	Maintain currency sign; change decimal separator to comma(,); insert thousand separator as dot(.); trim off leading zeros	€?(0+) €\-(0+)\ -\ ? (0+) (\.)\{1\}\ G\{€?\ -\ ? (\d\{1,3\}\}(?=(?:\d\d\d )+(?!d))	(?1€)(?2€-)(?3- ) (?4,)(?5\$&.)	/g (Global)
00000000.234 to 0,234	Trim off leading zeros; change decimal separator to comma(,)	(0)\{1,\}(\.)	0,	/g (Global)
000000001.234 to 1,234	Trim off leading zeros; change decimal separator to comma(,)	(\.)0+(\.)\{1\}\ ^0+	(?1,)(?20,)(?3)	/g (Global)



Post-Conversion	Description	Source (RegEx)	Target (Format)	Flag (Mode)
9999999999 to (999)999-9999	This requires two separate, sequential post-conversions	Post-conversion 1: \G^\(d{1})\ G(d{5}) Post-conversion 2: \G(\)+(d{3})	Post-conversion 1: (?1\(\$&)(?2\$&-) Post-conversion 2: (?1\$&))	
10/14/2002 to October 14, 2002		%m/%d/%Y	%B %d, %Y	
18 julio 2002 or 20 agosto 2002 or 05 mayo 2001 to 18-07-2002 or 20-08-2002	conversions for dates in languages other than English and French require two separate, sequential post-conversions	Post-conversion 1 (RegEx): (?:enero) (?:febrero) (?:marzo) (?:abril) (?:mayo) (?:junio) (?:julio) (?:agosto) (?:septiembre) (?:octubre) (?:noviembre) (?:diciembre) Post-conversion 2 (Date conversion): %d %m %Y	Post-conversion 1 (RegEx): (?101) (?202) (?303) (?404) (?505) (?606) (?707) (?808) (?909) (?1010) (?1111) (?1212) Post-conversion 2 (Date conversion): %d-%m-%Y	/g (Global)

### Explanation of First Example

The first example in the table above shows that to convert 1,234,567.89 to 1.234.567,89, the post-conversion you specify must change the dot(.) to a comma(,) and the comma(,) to a dot(.).

**Source** - The regular expression looks for exactly one occurrence of dot(.) or a comma(,) in the source data:

`(\\..) {1} | (\\,)`

**Target** - The target format specifies that if condition 1 is satisfied, replace the dot with a comma; else if condition 2 is satisfied, replace the comma with a dot:

`(?1,)(?2.)`

**Flag (mode)** - The /g or Global flag applies the formatting information globally (throughout the input string).

## Explanation of Second Example

The second example in the table above shows that to convert 1234567.89 to 1.234.567,89, the post-conversion you specify must search for and insert the thousand separators as dot(.) and the decimal separator as comma(,).

**Source** - The regular expression looks for exactly one occurrence of a dot(.) or sets the anchor to the beginning, if a digit between exactly one to three occurrences, look ahead three digits followed by not a digit:

```
(\.) {1} | \G(\d{1,3}) (?= (? : \d\d\d) + (? ! \d) )
```

**Target** - The target format specifies that if condition 1 is satisfied, then replace the dot with a comma, else if condition 2 is satisfied, then insert a dot(,):

```
(?1, ) (?2$&.)
```

**Flag (mode)** - The /g or Global flag applies the formatting information globally (throughout the input string).

# Index

## A

- anchor, 68, 105, 122
- Application, 49, 57, 64, 66, 74, 75, 77, 86, 101, 102
- Application Logic File, 66, 74, 75
- Application Tree, 77
- attributes, 14, 16, 47, 48, 49, 53, 56, 94
- AutoIndexVolAccept, 50, 57

## C

- CDA, 81, 83, 84, 85, 86
- CDA client, 84, 85
- Checking .log files for errors, 12
- Checking elements and dynamic fields with Simulator, 70, 71, 72, 73, 77
- com.edocs.app.user, 78

- Command Center, 15, 47, 54, 60, 66, 67, 78, 109
- Composer, 10, 65, 66, 67, 74, 75, 76, 77
- Configuring, 50, 57
- Configuring tasks
  - IXLoader, 56
  - StatementScanner, 27
- Content Access, 78
- Customer Self Service, 7

## D

- Data Definition File, 15, 16, 47, 48, 64, 65, 66, 67, 68, 70, 73, 75, 76, 86, 103, 104, 105, 106, 107, 110, 115, 116, 118, 119
- define end marker, 68, 104
- Definition Tree, 66, 68, 70, 73, 76, 104, 105, 107

DefTool, 10, 15, 16, 63,  
65, 67, 68, 69, 70, 71,  
72, 73, 74, 103, 104,  
105, 107, 109, 110,  
112, 113, 114, 115,  
116, 117, 118, 119

Development Tools,  
eaSuite, 10

documentation, 10, 11

## E

eaAssist, 9

eaDirect, 1, 2, 9, 10, 11,  
13, 14, 15, 47, 48, 49,  
50, 51, 54, 55, 56, 57,  
58, 60, 61, 62, 63, 64,  
65, 66, 67, 68, 74, 78,  
79, 82, 83, 86, 102,  
109, 110, 114, 115,  
119

eaMarket, 2, 9, 11

eaPay, 2, 10, 11

eaPost, 2, 10, 11

eaSample, 100

eaService, 2

eaSuite, 2, 7, 11, 81, 83

eaTraining, 64

edx\_common.jar, 84

**Email Notification**, 47,  
48, 57, 58, 59, 60, 61,  
62

extraction, 47, 49, 63, 65,  
70, 71, 72, 73, 109,  
110, 115

## F

Failed jobs, 35, 40

filtering, 77

## G

getDocumentReader, 79

Groups, 16, 66, 67, 74,  
76, 103, 105

## H

Help, 10, 11

documentation, 10

technical support, 11

HTML template, 47, 48

HTML templates, 47, 48

HTML view, 48, 64

## I

Index Volume Status, 27

Indexer, 47, 63, 66, 67,  
83, 86, 109

indexing, 15, 63, 64, 65,  
66, 67, 74, 78, 79

interface, 15, 79, 81, 83,  
101

internal anchor, 105

**IVNScanner**, 57, 61

IVNScanner task, 61

IXLoader task, 56

## J

job configuration, 54, 55,  
56, 60, 61, 62, 66

job schedule, 55

Jobs, 52, 59

failed, 35, 40

status, 34

Type, 34

**L**

Last Run, 34  
 link, 63, 97, 98, 99, 100  
 Logging into the  
   Command Center, 82,  
   89, 93  
 Login, 84  
 logs, 87, 89

**M**

MailNotification, 57, 62  
 Main Console, 51, 55, 57,  
   61  
 Marker, 68, 107  
 META-INF, 84

**N**

Next Run, 34  
 No operation, 35

**P**

Page style, 16, 103, 104  
 Parameters, 79  
 Password, 96  
 Production, 11, 56, 57,  
   61, 62, 66, 78  
 Purge App, 78

**R**

Recurring, 35  
 Regular expression, 109,  
   110, 113, 114, 120,  
   121  
 Reprocessing, 35  
 Run Time, 34

**S**

Scanner, 50, 55  
 SDK, 10  
 See Application Logic  
   File, 47, 48, 64, 65, 66,  
   67, 74, 75, 76, 77, 86  
 Sorting, 77  
 StatementScanner task  
   Report job, 27  
 Status, 34

**T**

Tables, 3, 16, 72, 103,  
   105  
 Template, 77  
 Type, 52, 59, 70

**U**

User, 65, 66, 67, 79, 84,  
   96, 97, 100, 110, 115,  
   119

**V**

Verify, 78, 79  
 View, 59, 66, 67, 74, 115,  
   116  
 Viewing, 100

**X**

XML, 14, 47, 48, 49, 50,  
   51, 52, 53, 55, 56, 57,  
   58, 59, 60, 61, 110,  
   116, 118  
 XSD, 14, 53  
 XSL, 48

