



Reporting Guide

ea**Care**

V4.0
Document ID: CRGN-05-4.0-01
Date Published: 8.11.03

© 1997–2003 edocs® Inc. All rights reserved.

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

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

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

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

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

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

Printed in the USA.

Table of Contents

- Preface 5**
 - About Customer Self-Service and eaSuite™ 5
 - About This Guide 8
 - Related Documentation 9
 - If You Need Help 9
- 1 About Reporting and Analytics 11**
- 2 Installation and Report Generation 13**
 - Installing and Configuring Reporting 13
 - Creating the Master Application 14
 - Creating an Indexer Job for the Master Application 14
 - Enrolling Users 18
 - Creating the Reporting Job 18
 - Configuring A Report Job 19
 - Indexing The Generated Report Data 22
 - Real-Time Reports 26
 - Upgrading from 3.4 28
- 3 Viewing Reports 29**
- 4 Production Considerations 33**
 - Performance Tuning 33

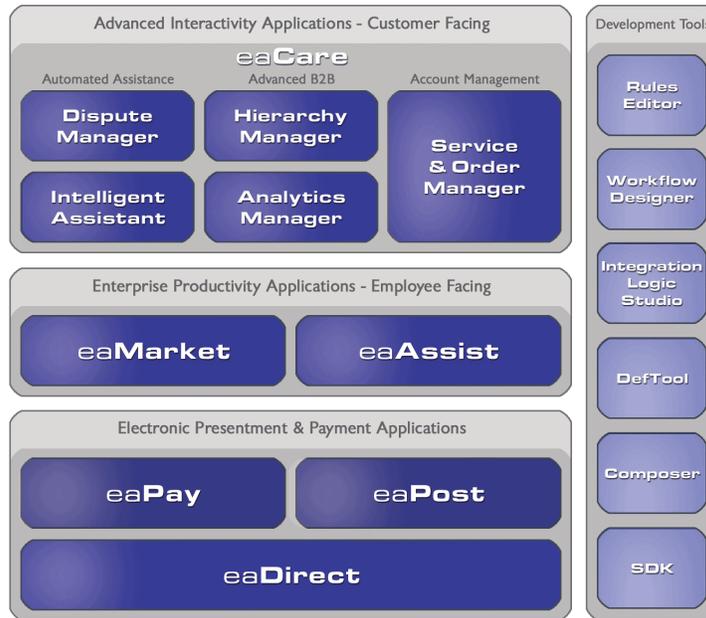
Preface

About Customer Self-Service and eaSuite™

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

eaSuite is comprised of four product families:

- Electronic Presentment and Payment (EPP) Applications
- Advanced Interactivity Applications
- Enterprise Productivity Applications
- Development Tools



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

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

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

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

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

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

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

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

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

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

About This Guide

This guide describes the tasks required to use the Analytics Module to create reports .

This guide is intended for the person that will install, configure and maintain reporting.

Related Documentation

Online Help and a PDF version of this guide are also available.

Online	How to Access
A PDF of this guide	A PDF of this guide is available on the eaDirect product 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:

Print Document	Description
<i>eaDirect Installation and Configuration Guide</i>	How to install eaDirect and configure it in a distributed environment.
<i>Data Presentation Production Guide</i>	How to set up and run a live eaDirect application in a J2EE environment.
<i>Deploying and Customizing J2EE Applications</i>	How to customize J2EE applications for deployment with the eaSuite.

If You Need Help

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

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

US Support Center

Natick, MA

Mon-Fri 8:30am – 8:00pm US EST

Telephone: 508-652-8400

Europe Support Center

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

Asia Pac Rim Support Center

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

Customer Central

<https://support.edocs.com>

Email Support

<mailto:support@edocs.com>

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

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

If the system crashed or hung, please tell us.

About Reporting and Analytics



The Reporting and Analytics module creates preconfigured telecommunication reports from live and/or 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

Report data is generated in two ways:

- The Report job creates reports for "large" statements
- Real-time report generation creates reports for "small" statements

The minimum size of large statements generated by the Report Job is defined in the reporting job configuration. The maximum size of small statements generated by real-time reports is defined in the report description XML file.

Both methods produce report data. The Report job creates data which is not viewable until it is indexed by the Indexer job. The real-time report generator creates reports for immediate viewing, and also appends each new report to a larger report file. The larger report file of real-time reports is indexed by the Indexer job, so a real-time report only needs to be created once.

Installation and Report Generation

2

Installing and Configuring Reporting

eaCare Reporting is installed when you installed eaDirect. The license key that you provide when installing determines which products are installed.

The features added to eaDirect when you install Reporting support creating indexed and/or dynamic reports. The reportSample EAR provides a way to view reports. If you haven't already installed *ear-reportSample.ear*, then do that first. Refer to the *eaDirect Installation and Configuration Guide* for your platform for more information about installing EAR files.

If you don't already have users defined, define at least one before running the Report job. If there are no defined users before you run the Report job, then the job will return a status of **no operation**. Also, the Report job only creates reports for enrolled users each time it runs.

The following sections describe how to create reports using the Report jobs and the provided supporting files. Note that the NationalWireless files provided for Reporting are not compatible with the NationalWireless files provided for other sample applications.

Creating the Master Application

The Master application is used when creating an Indexer job to index the data for reporting, and publish the files for the Indexer job.

To Create the Master Application:

1. Create a new (master) application to index the sample data named **NW**.
2. Use `/edx/reportSample/ejb/ReportDataSource` as the datasource when creating the new application.
3. Chose the appropriate value for "Index Partition Count". See the *Data Presentation Production Guide* for more information about this parameter.

Creating an Indexer Job for the Master Application

Create an Indexer job under the Master application, and publish the files required by that job. These examples call the Indexer job "Indexer".

To Publish the Indexer Files:

1. Click **Launch Publisher**, and then **Create**.

Select a Version Set Type

Dynamic Web Views

Job Type	Number of Auxiliary Files
HTML	0 1 2 3 more
XS	0 1 2 3 more
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 more
HTML Output	0 1 2 3 more
Indexer	0
Report	0
XML Email Notification	0
XML Loader	0
XML Output	0

- 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:

3. Select the application name from the drop-down list. For example, **NW**.
4. Browse and select the DDF file for the Indexer job. The sample DDF file is *\$EDX_HOME/samples/NW_Report/MasterDDN/IndexerJob/NWIndexer.ddf*.
5. Click **submit**. Publisher displays the Submission screen with details about the DDF file:

<p>Submission</p> <p>Application: NW View Type: DOC_CONFIG View Name: Indexer Timestamp: Tue Jul 08 13:57:06 GMT-05:00 2003</p> <p>This version set contains the following files:</p> <p><u>NW/DOC_CONFIG/Indexer/20030708135706/NWIndexer.ddf</u></p>
--

6. Close the Publisher window.

Creating and Configuring the Master Indexer job

1. At the Create New Job screen in Command Center, click **Configure Job and Continue**. eaDirect displays the job configuration screen:

Submit Changes and Schedule		Refresh	Reset	Help
Task 1: Scanner				
Input File Path:	/opt/EDCSbd/Input/NW/			
Input File Name:	*.txt			
Output File Path:	/opt/EDCSbd/Data/NW/			
Task 2: Indexer				
DDF Path:	/opt/EDCSbd/AppProfiles/NW/DOC_CONFIG/Indexer/20090708135706/NWIndexer.ddf			
Task 3: IXLoader				
Load Method:	Direct			
Task 4: AutoIndexVolAccept				
Action on Index Volume:	Auto Accept			

- When finished entering configuration parameters, click **Submit Changes and Schedule**. eaDirect asks “OK to submit this configuration?”
- Click **OK**. eaDirect submits the job configuration parameters and displays the Schedule screen. Do not schedule the Indexer job at this time.
- Be sure to copy the *NationalWireless.txt* file to the input directory before running the Indexer job. For example:

```
cd /opt/EDCSbd/samples/NW_Report/MasterDDN/datafile
cp NationalWireless.txt /opt/EDCSbd/Input/NW
```

Caution The *NationalWireless.txt* file in the *NW_Report* directory is NOT the same as the *NationalWireless.txt* file in the other *Sample* directories.

- Run the Indexer job.

Enrolling Users

Users must be enrolled before running the Reporting job. Log on to the reportSample application and click on Enroll Now. For example, to bring up the sample application on a WebLogic server:

```
http://audi:7001/reportSample
```

Creating the Reporting Job

Create a Report job under the Master application, and publish the files required by that job.

To Publish Files for a Report Job:

1. Create a second job for your reporting 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:

DDF File for Dynamic XML:

Browse to find the sample report description XML file, select it, and click Submit to publish the XML file as a version set. For example, */opt/EDCSbd/samples/NW_Report/MasterDDN/ReportJob/NWReportDescription.xml*.

Browse to find the sample DDF file for XML output, select it, and click Submit to publish the XML file as part of the same version set. For example, */opt/EDCSbd/samples/NW_Report/MasterDDN/ReportJob/NWDynXML.ddf*.

3. Close the Publisher browser window.

Configuring A Report Job

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

Application: NW Job: Report	
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 fields or for Help click the appropriate button at the top of the screen.	
<input type="button" value="Submit Changes and Schedule"/> <input type="button" value="Refresh"/> <input type="button" value="Reset"/> <input type="button" value="Help"/>	
Task 1: StatementScanner	
Index Volume Status:	Accepted ▾
Scan IVN Starting From (Number of Days):	7
Account Resolver:	edx/reportSample/ejb/CDAAccountResolv
Account Attribute:	uid
Byte Count Threshold:	0
Queue Size:	100 ▾
Task 2: ReportTask	
Report File Output Directory:	/opt/EDCSbd/Input/NW_rpt/
Max Number of Manual Retries:	1 ▾
Task 3: ReportCollector	
Output File Path:	No user configuration necessary

Task 1: StatementScanner

The Statement Scanner scans the index volume table to create a list of statements (documents) being indexed and passes the doc id of the document to the Report task if the account is valid (enrolled).

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.				
	<table border="1"> <tr> <td>Accepted</td> <td>(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. 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.</td> </tr> <tr> <td>Rejected</td> <td>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). 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.</td> </tr> </table>	Accepted	(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. 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.	Rejected	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). 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.
Accepted	(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. 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.				
Rejected	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). 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.				
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, <code>edx/reportSample/ejb/CDAAccountResolver</code> .				
Account Attribute	For the CDA Account Resolver, use <code>uid</code> .				
Byte Count Threshold	Enter the size in bytes that will determine when not to process this statement. Statements below this size will be processed by the real-time report processor, not by the report job. The default (0) means that reports will be generated for all statements, and JIT reporting will not be used.				
Queue Size	Specifies how many statements will be scanned in the queue for each concurrent task to process.				

Task 2: ReportTask

The Report task uses the published report description XML file to extract data from the files that were created by running the Indexer job. It then writes the new report files.

ReportTask Configuration (Report job)	
Field	What to enter
Report File Output Directory	The path to the directory where the Report job writes the reports.
Max Number of Manual Retries	Determines how many times a failed statement is retried. After this number, a failed statement is marked as failed.

Task : ReportCollector

The ReportCollector task generates one file for each indexed data file.

No configuration is necessary for this task.

Submit and schedule the job. Be sure you have enabled concurrency for the eaDirect scheduler.



Caution

There must be at least one user enrolled to the application before running the Report job for it to run successfully. Otherwise, the job will report a No Operation in the Statement Scanner.

Report Job Results

The report job uses the master doc date of the statement to retrieve the version set of RDX and Dynamic XML DDF files when it generates and parses the XML statement produced by the statement scanner.

One report file is generated for each indexed data file, which contains all the reports defined in RDX for all statements in that data file.

After an IVN is scanned and processed by a report job, it will not be scanned again even if you run another report job under the same DDN.

If you get a No Operation status from the report job, the reason can be:

- There are no new IVNs, or,
- There are new IVNs, but there are no users enrolled. In this case, the default CDAAccountResolver returns false for each account. or,
- There are new IVNs and enrolled users, but the byte counts of all statements are less than the defined byte count threshold setting.

Indexing The Generated Report Data

The data generated by the Report job must be indexed so that customers can view the reports. This indexer job also indexes the data created by the JITRptCollector job. You can use the sample DDFs with your ALF and HTML files to index the report data that was generated by the Report job. As part of creating the Indexer job, publish Html views for each report, and then create and run Indexer job.

The Reporting job creates one report file for each data file. The file name format is “r_IVN_<DataFileName>.txt”, where IVN is the IVN number of indexed data file and <DataFileName> is the name of the indexed data file.

To Create the Reporting Application:

Create a new application, using the name of the Master application with "_rpt" appended to the name. For example, if the master application is "NW", then the reporting application must be "NW_rpt".

Use `/edx/reportSample/ejb/ReportDataSource` as the datasource when creating the new application.

To Publish the Html Views:

Start the Publisher, click on **Create**, and click on **1** under Number of Auxiliary Files for the HTML job type for each Html view (except for CostSumByType, TopNExpCalls, TopNExpNumCalled for which you must click on **2**).

All the sample files are located in *\$EDX_HOME/samples/NW_Report/ReportDDN/HtmlViews* (for Unix, *%EDX_HOME%\samples\NW_Report\ReportDDN\HtmlViews* for Windows), where the subdirectory for each view name matches the view name. The following list shows the sample view names and the associated files that you must publish:

Html View Name: CostSumByMonth
DDF: *CostSumByMonth.ddf*
ALF: *CostSumByMonth.alf*
HTML template: *CostSumByMonth.htm*
Auxiliary files: *SubReport.htm*

Html View Name: CostSumByTime
DDF: *CostSumByTime.ddf*
ALF: *CostSumByTime.alf*
HTML template: *CostSumByTime.htm*
Auxiliary files: *SubReport.htm*

Html View Name: CostSumByType
DDF: *CostSumByType.ddf*
ALF: *CostSumByType.alf*
HTML template: *CostSumByType.htm*
Auxiliary files: *SubReport.htm, GroupWithDynPat.htm*

Html View Name: TopNCtryCalled
DDF: *TopNCtryCalled.ddf*
ALF: *TopNCtryCalled.alf*
HTML template: *TopNCtryCalled.htm*
Auxiliary files: *SubReport.htm*

Html View Name: TopNExpCalls
DDF: *TopNExpCalls.ddf*
ALF: *TopNExpCalls.alf*
HTML template: *TopNExpCalls.htm*
Auxiliary files: *SubReport.htm, GroupWithDynPat.htm*

Html View Name: TopNExpNumCalled
DDF: *TopNExpNumCalled.ddf*
ALF: *TopNExpNumCalled.alf*
HTML template: *TopNExpNumCalled.htm*
Auxiliary files: *SubReport.htm, GroupWithDynPat.htm*

Html View Name: TopNLongestCalls
DDF: *TopNLongestCalls.ddf*
ALF: *TopNLongestCalls.alf*
HTML template: *TopNLongestCalls.htm*
Auxiliary files: *SubReport.htm*

Html View Name: TopNNumCalled
DDF: *TopNNumCalled.ddf*
ALF: *TopNNumCalled.alf*
HTML template: *TopNNumCalled.htm*
Auxiliary files: *SubReport.htm*

Creating the Reporting Indexer Job

The following steps show how to create an Indexer job to index the generated report data:

1. Create an Indexer job for the reporting application.
2. Start the Publisher, and create the following version set for the report indexer job:

Resource	Value
DDF	<i>/opt/EDCSbd/samples/NW_Report/ReportDDN/IndexerJob/ReportIndexer.ddf</i>

The indexer job should index all the files under its input directory.

The Z_DOC_DATE of indexed report is the one from the Z_DOC_DATE of report file, which is actually the doc date of the original statement. The Z_PRIMARY_KEY of indexed report is the master doc id of original statement.

3. Close the Publisher, return to the Indexer job and check the following information:

Resource	Value
Input File Path	<i>/opt/EDCSbd/Input/NW_rpt</i>
Input File Name	<i>*.txt</i>
Output File Path	Keep the default

4. Save the job and schedule it.

Real-Time Reports

The real-time reporting feature creates reports for data that is smaller in size than the value configured in the reporting job. After creating the report, it saves it in a file. The JIIRptCollector job reads the real-time reports, concatenates them into a single report file to be indexed by the indexer job, and deletes the real-time files that have been indexed. After indexing, the next request for that report will be retrieved from the index instead of in real-time.

Real Time (or Just-In-Time) reporting is enabled by entries in the JITConfig section of the RDX file, which defines the maximum byte count for a JIT report, and the output directory of the *.jit* file .

When a user views a JIT report, a corresponding *.jit* file is generated in the output directory defined in the JITConfig section of the RDX. If the directory is undefined, the default is */EDCSbd/Output/<masterDDN>_rpt*.

After a number of *.jit* files have been generated, run the JIIRptCollector job, which cleans up the directory by creating a single report file from the saved real-time reports, along with a *.lst* file containing the names of the collected *.jit* files. After the new report file has been indexed by the reporting application's index job, the next run of the JIIRptCollector job deletes the *.jit* files and the associated *.lst* file. Note that JIT reporting does not require running the JIIRptCollector job, but you should watch the directory to make sure it does not fill up with JIT files, and that the number of JIT files doesn't slow down JIT report retrieval.



If the "Ignore statements whose byte counts are less than" value in the Report Job configuration is greater than the JIT byte count threshold defined in the RDX (JITConfig), the reports for statements with byte counts within these two values will never be generated.

To create the JItrptCollector job:

1. Create a JIT Report Collector job for the reporting application.

Task 1: JItrptCollector	
JIT Report File Directory:	/opt/EDCSbd/Output/NW
Collected Report File Directory:	/opt/EDCSbd/Input/NW
Collection Threshold:	100

Check the following information:

Resource	Value
JIT Report File Directory	Defines where JIT report files are generated. This must match the value in the reporting RDX file.
Collected Report File Directory	Defines where the JItrptCollector job puts the collected JIT reports. It must match the directory where the report Indexer job looks for input.
Collection Threshold	Defines how many JIT files must be in the JIT Report File Directory for the JItrptCollector job to run.

2. Submit and schedule the job. Be sure you have enabled concurrency for the eaDirect scheduler.

The JIT Report Collector job reports a NoOp status if:

- There are no *.jit files to collect.
- The number of JIT files is less than the value of JIT Report Collector parameter “Don't collect if number of JIT report files less than”.

After running the JItrptCollector job, be sure to run the reporting application's Indexer job. This makes the collected JIT files available to users, and let's the JItrptCollector job know that it is safe to delete the collected JIT files the next time the job runs.

Upgrading from 3.4

If you are upgrading from eaCare Reporting 3.4, then follow these guidelines, referring to this document for details:

1. Update eaDirect as described in the *eaDirect Installation and Configuration Guide*.
2. Republish the report description XML file and DDF file for XML output under the Master DDN.



eaDirect's Indexer job allowed you to specify which fields you wanted to index in the job for previous versions. As of 4.0, the DDF file specifies which fields will be indexed. You will need to update your DDF files as part of upgrading eaDirect to 4.0, and rerun the Master indexer job.

3. Create and configure a new Report job.
4. Create and configure a new Report DDN.
5. Create a new Indexer job to index the reports, and publish the new views for each report type. Version 3.4 used 8 indexer jobs, which can be deleted.
6. Schedule and Run the master DDN indexer job to index new files and run report job to generate reports. Wait at least one day after the last 3.4 statement is indexed before running the Master Indexer.

Viewing Reports

3

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. These examples use the account number 4694878 from *NW_Report/MasterDDN/datafile/NationalWireless.txt*.

1. Open reportSample and choose the master application, then log on using one of the accounts from the file that you indexed for the Report job. The URL is *http://host:port/reportSample*. For example:

http://myHost:7001/reportSample

Use the following URL format

```
http://<host>[:port]/reportSample/User?app=UserMain&jsp=/user/jsp/HistoryList.jsp&ddn=<masterDDN>
```

to accomplish the following tasks:

1. Display a login screen for the user to log in or enroll.
2. Display account summary/history.
3. Display reports.

For Example... Select the master DDN which is used to index the original data file. Before you can view reports by using reportSample for a master DDN, you need to create its corresponding report DDN with name as masterDDN_rpt and publish related files.

Master DDN:	NW
-------------	----

Submit Query Reset

2. Select the Master DDN (NW in our example).

Enroll Now

User Login... Enter your username, password and click "Submit." If you do not have a username or password, [Enroll Now](#) to sign up for your electronic bill.

Username:	user1
Password:	*****

Submit Reset

- Log on as the one of the users you created before running the Master Indexer.

The screenshot shows the edocs ONLINE ACCOUNT MANAGEMENT & BILLING interface. At the top, there is a navigation bar with "Account Summary", "Edit Profile", and "Logout" links. Below this is the "Account History" section, which includes a table with the following data:

View	Reports	Z_DOC_DATE	CustName	StatementDate	DueDate	AmtDue	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_10
	Report	06-11-2003	CHAOS CLEARED TECHNICAL HELP	MARCH 25, 2001	NO AMT DUE	1225.50CR										

- Click on the Report link to view the reports.

The screenshot shows the edocs ONLINE ACCOUNT MANAGEMENT & BILLING interface with the "Report" form. The form includes the following fields and instructions:

- Select a call type.** This call type is actually the groupKey of groupCriteria defined for a report in RDX. For telecom users, the groupKey is usually, but not limited to, call type. Select from drop-down list if you are using NationalWireless test data or type your own if not. When type your own, make sure the value you type in is one of the valid values for the groupKey.
- Select NationalWireless call type:** (dropdown menu)
- Or type your own:**
- Select a report type defined in RDX.** Each report is viewed as a HTML View whose name must be the same as the report id defined for that report type in RDX. Use the drop down list to select a report type (view) or type you own view name if you are using different report ids/view names.
- Select default report/view name:** (dropdown menu)
- Or type your own:**
- submit** button

Select the call type from the first drop-down and then the default report/view name from the second drop-down to view the reports. For example, select:

Parameter	Value
Call Type	All Call Type
Report/View Name	TopNExpCalls

Which will create a report similar to this:

Report Type - TopNExpCalls							
Details and Totals							
Top N Expensive Calls - Details							
LineCallDate	LineCallTime	LineNumCalled	LineCallCity	LineCallState	LineCallType	LineCallMins	LineCallUsage
01/30	12:12	603 764-2976	NASHUA	NH	D	75.2	25.14
02/01	11:35	970 408-3234	STAMFORD	CT	D	22.8	10.61
02/04	04:02	512 438-7956	MEDIA	PA	D	15.5	7.58
01/31	01:41	542 456-3681	PORTLAND	OR	D	12.3	6.98
01/30	10:55	978 401-8279	READING	WY	D	5.0	2.90
01/28	07:57	970 408-3234	STAMFORD	CT	E	5.0	2.35
01/28	06:02	352 305-1209	LEHIGH	OH	E	2.8	1.07
Top N Total							
TotalCount	TotalDuration	TotalAmount					
7	138.6	56.63					

Performance Tuning

The Reporting module, including the Report job, JIT reporting and JIT Report Collector job, need to be tuned properly to achieve best performance. Follow these guidelines to achieve the best performance of reporting:

Byte Count Threshold

This Report job parameter determines whether a statement's reports are generated at batch time or at web access time (JIT). Choose a parameter that yields reasonable JIT report response time in your deployment under normal load. If the value is too low, it may put too much burden on the report job, and the job may take too long to finish. If it is too high, the response time of JIT reports may be too long for a user to wait. To start testing, find the byte count of a document with a few thousand (call) details. Check the response time of JIT reports during normal load. Reduce the byte count threshold if the response is too slow, or increase it if the response is within acceptable values.

It's important to design a testing module with normal load. Include the number of concurrent users, how many may view reports, and how many may view the other reports of the same statement (the second visit is significant faster than the first because there is no need to generate a JIT file.)

To tune this parameter, run the master DDN indexer job, then view reports for the test statements without running the report job. Use typical data files for JIT testing.

Queue size

This Report job parameter defines how many doc ids are sent to the next report task for each report job instance. If the value is too low, the number of job instances will be very high and the overhead introduced by PWC can be costly. If the value is too high, the load may not be balanced properly, and in extreme cases, only one job instance will process all the documents, which eliminates the benefits of parallel processing provided by statement-level processing.

For example, assume:

- the average number of statements in a data file is M
- $k\%$ of the statements are above the JIT byte count threshold
- the accounts are resolved as true by the account resolver
- the number of concurrent threads in the PWC scheduler is n

Then the queue size Q should be:

$$Q \leq (M * k\%) / n$$

This formula helps ensure that all the concurrent job instances are utilized.

If the statements in the data file are fairly even in size, start from a number close to the upper limit as previously defined. If the statements have vastly different sizes, start from a number far below the upper limit, approximately 10 times lower.

Using a low number (for example, 1) is not recommended unless the number of statements in the data file is very low.

If you cannot estimate a value using the preceding information, start from the default 100. Watch the CPU usage on the application server and database server while the report job runs. Increase the number if the application server is under-used, but the database server is over-used, and vice versa.

JIT byte count upper limit

This value is defined in the RDX as “MaxByteCount” under the “JITConfig” tag. It has less impact on performance than the first two parameters.

Usually, a data file is indexed, then the report file is generated and indexed. If the user comes in after that, JIT reporting is only be invoked for statements whose byte counts are less than the defined byte count threshold.

However, there are cases where JIT is invoked for statements whose byte counts are larger than the threshold. See the section *JIT Considerations* for more information. Do we want to generate JIT report for a statement with 1 million details in it? The answer is no, because the browser eventually times out, so it wastes time to create a JIT report.

To avoid this, you can set the upper limit. Set this limit to a value such that the time to generate a JIT statement with this byte count is longer than a user will wait. When tuning this parameter, remember to remove all JIT files before accessing reports, or else you may pull up an existing JIT file, and your assertion will be wrong.

JIT Report Collect job, “Collection Threshold”

This parameter affects how often the JIT Report Collector job runs. Setting this value too low generates too many job instances and too many small report files which puts a burden on the file system. Setting this value too high may end up flooding the directories with too many JIT files.

The default value 100 should be adequate. Watch your file system usage during JIT testing while the job is running and make adjustments, if necessary.

JIT Considerations

JIT reporting introduces a few interesting side effects you need to be aware.

- **New users** - When a new user enrolls, there are no indexed statements for that user. So if the user immediately views a report, JIT is used. This is fine for small statements, but for large statements users may get a timeout or an error.
- **Duplicate reports** - A report may be generated twice for the following case: a data file is indexed, but before the report job and report DDN indexer jobs finish, a user views a report.

In this case, JIT reports are generated, and both the JIT and Report job copies are indexed. This is not a problem viewing a report from the UI, which just chooses one of the reports. Since this case does not occur very often, and report files are small, the extra disk space introduced by this duplication should not be a concern.

- **Doc date** - Note that the Report doc date is not the date when the statement was indexed, it is the master doc date. This is required to support version control.

Caching

This section describes how the caching works in the report job.

To improve performance, the ReportTask of the Report job uses caching. The content of the RDX is cached to avoid multiple file accesses. The cache may become stale if the published RDX file is modified directly. Re-publishing an RDX file does not cause a problem because the cache automatically detects modification of the RDX and re-loads the file.