



# EngageOne Digital Self Service

Document Version 604-00002-001

## Operations Guide

US English Edition

---

Copyright ©2017 Pitney Bowes All rights reserved.

This publication and the software described in its supplied under license and may only be used or copied in accordance with the terms of such license. The information in this publication is provided for information only, is subject to change without notice, and should not be construed as a commitment by Pitney Bowes Software. Inc. (PB). To the fullest extent permitted by applicable laws PB excludes all warranties, representations and undertakings (express or implied) in relation to this publication and assumes no liability or responsibility for any errors or inaccuracies that may appear in this publication and shall not be liable for loss or damage of any kind arising from its use.

Except as permitted by such license, reproduction or any part of this publication by mechanical, electronic, recording means or otherwise, including fax transmission, without the express permission of PB is prohibited to the fullest extent permitted by applicable laws.

Nothing in this notice shall limit or exclude PB liability in respect of fraud or for death or personal injury arising from its negligence. Statutory rights of the user, if any, are unaffected.

\*TALO Hyphenators and Spellers are used. Developed by TALO B.V., Bussum, Netherlands

Copyright © 1998 \* TALO B.V., Bussum, NL

\*TALO is a registered trademark ®

Encryption algorithms licensed from Unisys Corp. under U.S. Patent No. 4, 558,302 and foreign counterparts

Security algorithms Copyright ©

1991 – 1992 RSA Data Security Inc.

Copyright © DL Technology Ltd 1992-2010

Barcode fonts Copyright © 1997 Terrapin Solutions Ltd. with NRB Systems Ltd.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Artifex and the Ghostscript logo are registered trademarks and the Artifex logo and Ghostscript are trademarks of Artifex Software, Inc.

Copyright (c) 2000 - 2015 The Legion of the Bouncy Castle Inc. (<http://www.bouncycastle.org>)

This product contains the Regexp++ library. Copyright © 1998-2000 by Dr. John Maddock

PostScript is a trademark of Adobe Systems Incorporated.

PCL is a trademark of Hewlett Packard Company.

ICU License - ICU 1.8.1 and later

Copyright (c) 1995-2006 International Business Machines Corporation and others.

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

Matra 0.8b (<http://matra.sourceforge.net/>) The contents of this documentation are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this documentation except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

Otherwise all product names are trademarks or registered trademarks of their respective holders.

---

## Table of Contents

1	Overview .....	1
1.1	Batch and Server Jobs .....	1
2	Jobs Run on the Batch Server .....	1
2.1	Billers to Business Jobs .....	1
2.1.1	Batch – Company Account Link Processor .....	1
2.1.2	Batch – Schema Synchronizer .....	2
2.2	Bill Loading Jobs .....	2
2.2.1	Batch – Bill Xfer .....	2
2.2.2	Batch – Business UBF Bill Load .....	3
2.2.3	Batch – Bill Purge .....	4
2.3	Notification Jobs .....	4
2.3.1	Batch – Bill Notify .....	4
2.3.2	Batch – Audit Purge .....	5
2.3.3	Batch – Notification Purge .....	6
2.4	Servers .....	7
2.4.1	Server – Notification DQ Server .....	7
2.4.2	Server – Notification IQ Server .....	8

## 1 Overview

The *EngageOne Operations Guide* provides information about batch, server, and token jobs. For each process, this document provides the launch directories and commands, a high-level process overview, a table inclusive of configuration properties, and insight into potential failure points.

### 1.1 Batch and Server Jobs

While batch jobs are specific to software tasks that perform certain functions (and run on a periodic basis until completion), servers include software tasks that respond to a service request and run until explicitly stopped. That noted, generally, batch jobs and servers are *scheduled* (running at designated dates/times) and do not require user intervention. Batch jobs typically integrate with job schedulers (such as Jenkins). Servers run continuously in the background, waiting for (and operating on) requests. These background servers perform functions to include delivering notifications, performing service management tasks, and processing payments. Batch and server jobs run on the batch server.

## 2 Jobs Run on the Batch Server

### 2.1 Biller to Business Jobs

#### 2.1.1 Batch – Company Account Link Processor

Launch Directory:	<install dir>/batch/company_account_link_processor
Launch Command:	ant run
Configuration Dir:	<install dir>/batch/company_account_link_processor/resources/
Configuration File(s):	company_account_link_processor.properties

##### 2.1.1.1 Overview

This job processes any changes in top-level organizational IDs assigned to a company definition.

The Agent portal allows an agent to add or remove organizational IDs from a company. This process may be time-consuming and, as such, is queued for operation rather than executed while the agent waits.

By default, this job runs every 5 minutes. You can schedule this job to run at any frequency contingent upon your overall system loads. It is important to remember that the delay between an agent action and the customer's ability to see the resulting update is controlled by the frequency at which you run this process.

##### 2.1.1.2 Configuration

company_account_link_processor.properties	
batch.success	ID of the batch job e-mail template. ("batch_success")
batch.failure	ID of the batch job e-mail template. ("batch_failure")

##### 2.1.1.3 Potential Failures/Errors

If this process fails, the database may not be accessible or a tablespace is full.

## 2.1.2 Batch – Schema Synchronizer

Launch Directory:	<install dir>/batch/bill_xfer
Launch Command:	ant run
Configuration Dir:	<install dir>/batch/bill_xfer/resources/
Configuration File(s):	schema_synchronizer.properties

### 2.1.2.1 Overview

This job processes any hierarchy changes in companies, resulting from the previous job (Company Account Link Processor).

By default, this job runs every 5 minutes. You can schedule this job to run at any frequency contingent upon your overall system loads. It is important to remember that the delay between an agent action and the customer's ability to see the resulting update is controlled by the frequency at which you run this process.

### 2.1.2.2 Configuration

schema_synchronizer.properties	
move	Set to TRUE indicates that input files will be transferred to a new directory, and then the database will be updated. Set to FALSE indicates that only the database will be updated.
move.method	This flag should be "RENAME", unless that fails to work in your environment. "COPY_AND_DELETE" will work in all environments but is slower and uses more disk space.

### 2.1.2.3 Potential Failures/Errors

If this process fails, the database may not be accessible or a tablespace is full.

## 2.2 Bill Loading Jobs

### 2.2.1 Batch – Bill Xfer

Launch Directory:	<install dir>/batch/bill_xfer
Launch Command:	ant run
Configuration Dir:	<install dir>/batch/bill_xfer/resources/
Configuration File(s):	xfer.properties

### 2.2.1.1 Overview

The bill transfer (Xfer) job moves files from an input directory into a bill processing directory so the bill loader jobs can process them. This job also creates these directories the first time it is run.

In some implementations, this job is augmented or replaced by an FTP job that pulls data from another system, places it in the input directory, and then renames it in the processing directory.

If this job is not run, then bill files will accumulate in the input directory and no bills will be loaded into the system. It is safe to run this job when there are no new bill files to be processed.

### 2.2.1.2 Configuration

<b>xfer.properties</b>	
move	Set to TRUE indicates that input files will be transferred to a new directory, and then the database will be updated. Set to FALSE indicates that only the database will be updated.
move.method	This flag should be "RENAME", unless that fails to work in your environment. "COPY_AND_DELETE" will work in all environments but is slower and uses more disk space.

### 2.2.1.3 Potential Failures/Errors

**File-related errors:** These errors are triggered when there is not enough disk space to process the requests.

**Database-related errors:** These errors are triggered when the job cannot access the database.

## 2.2.2 Batch – Business UBF Bill Load

Launch Directory:	<install dir>/batch/bill_loader_business_ubf
Launch Command:	ant run
Configuration Dir:	<install dir>/batch/bill_loader_business_ubf/resources/
Configuration File(s):	BusinessBillLoaderJob.properties

### 2.2.2.1 Overview

This process loads bills into the database for bill presentment. Each instance of this job processes a single bill file containing one or more (and, commonly, many) bills.

You can schedule this job in one of two ways: (1) Run periodically (and coordinated with bill production schedules), or (2) Run based on file arrival (Jenkins includes a plug-in). If no bill files are available for processing, this job will still return SUCCESS; therefore, it is safe to schedule this job when there are no new bill files to be processed.

This job is a good candidate for configuring multiple simultaneous instances (particularly, if you need scalability and have several bill files arriving at the same time).

### 2.2.2.2 Configuration

<b>BusinessBillLoaderJob.properties</b>	
data.compress	This property represents the flag set to control compression of the bills in the database. For production systems, it is recommended that this flag is set to TRUE.
loaded.status	One of two values: LOADED – Use this status if you want to manually publish bills. PUBLISHED – Bills will be immediately available for viewing by users.
batch.success	ID of the batch job e-mail template. (" <b>batch_success</b> ")
batch.failure	ID of the batch job e-mail template. (" <b>batch_failure</b> ")
batch.noop	ID of the batch job e-mail template. (" <b>batch_noop</b> ")
notification.onNoop.enabled	This property represents the flag that is set to enable notifications when a job runs, but there is no data to process.

## BusinessBillLoaderJob.properties

notification.onNoop.emails	This property represents the e-mail address to send messages when there is no data to process.
----------------------------	--

### 2.2.2.3 Potential Failures/Errors

**File System-Related Failures** – To avoid these failures, ensure you run the bill\_xfer job prior to running this process and that it copied your input files into the proper directory; in addition, ensure there is adequate disk space – as the ETL files can become sizeable.

**Input File Errors** – This error type may be triggered if data corruption occurred during transfer or a file was improperly formatted by the system that generated it. Unexpected file format changes can cause failures because teams responsible for upstream systems may not realize the impact their changes can have on bill file loads.

**Database Errors** - Many IT organizations configure Oracle table spaces to be *fixed* in size. As the bill history and the datamart grow, these table spaces can fill (and do not extend automatically). If this process fails, a tablespace is full or the database may not be accessible.

### 2.2.3 Batch – Bill Purge

Launch Directory:	<install dir>/batch/bill_purge
Launch Command:	ant run
Configuration Dir:	<install dir>/batch/bill_purge/resources
Configuration File(s):	purge.properties

#### 2.2.3.1 Overview

The bill purge job removes old bills that are beyond a certain cut-off date; in addition, the bill purge job removes expired or rejected bills.

#### 2.2.3.2 Configuration

purge.properties	
purge.exp.rej.only	This property specifies if the bill data purge job will delete bill data for files older than {purge.older.than.Xdays} – and in an expired or rejected status – or if it will delete all files older than {purge.older.than.Xdays}.
batch.success	ID of the batch job e-mail template. (“batch_success”)
batch.failure	ID of the batch job e-mail template. (“batch_failure”)

## 2.3 Notification Jobs

### 2.3.1 Batch – Bill Notify

Launch Directory:	<install dir>/batch/bill_notify
Launch Command:	ant run
Configuration Dir:	<install dir>/batch/bill_notify/resources/
Configuration File(s):	Bill_notify.properties

### 2.3.1.1 Overview

This job sends notifications to registered users when they have received one or more new bills for accounts with which they have access.

This job is usually scheduled to run nightly (although it may be scheduled to run immediately after a bill load job). By running the job nightly, users (with access to several accounts for which there are new bills) have the advantage of receiving a single notification for all their accounts. Notifications are consolidated per run (only).

If no new bills have been loaded into the system, it will exit without performing any operations.

If the job fails to run, users will get the pending notification the next time the job runs successfully.

### 2.3.1.2 Configuration

bill_notify.properties	
b2b.few.accts.msg.id	ID of the batch job e-mail template for a user with access to few accounts. ("bill_new_doc_few_accounts")
b2b.many.accts.msg.id	ID of the batch job e-mail template for a user with access to many accounts. ("bill_new_doc_many_accounts")
b2c.accts.msg.id	ID of the batch job e-mail template for a user with access to a single account. ("bill_new_doc_one_account")
batch.success	ID of the batch job e-mail template. ("batch_success")
batch.failure	ID of the batch job e-mail template. ("batch_failure")
batch.noop	ID of the batch job e-mail template. ("batch_noop")
notification.onNoop.enabled	This property represents the flag that is set to enable notifications when a job runs, but there is no data to process.
notification.onNoop.emails	This property represents the e-mail address to send messages when there is no data to process.

### 2.3.1.3 Potential Failures/Errors

Database-related errors – This job will fail if the database is inaccessible.

## 2.3.2 Batch – Audit Purge

Launch Directory:	<install dir>/batch/audit_purge
Launch Command:	ant run
Configuration Dir:	<install dir>/batch/audit_purge/resources/ <install dir>/config/common/
Configuration File(s):	audit_purge.properties file_path.properties user_config.properties

### 2.3.2.1 Overview

This purge job deletes records from the AUD\_EVENT table; specifically, this job deletes records that exceed (and, therefore, are older) than the designated number of days to keep audit data in this table. This job also creates an audit log file in the dir.logs location.



*Audit export* is a subset of the audit purge job. Audit export writes all selected audit data to be purged to the `<install_dir>/logs/auditEvents_yyyyMMddhhmmss.log` file. After this log file is created successfully, the audit purge job then deletes these records from the **AUD\_EVENT** table. In short, before select audit data is purged, it is written to disk via audit export.

### 2.3.2.2 Configuration

audit_purge.properties	
job.name	Job name used in the job status notification messages.
purge.chunk.size	This property represents the number of records to be deleted at a time; size of delete chunk.
batch.success	ID of the batch job e-mail template. ("batch_success")
batch.failure	ID of the batch job e-mail template. ("batch_failure")

file_path.properties	
dir.logs	Directory to look for (and which houses) the audit logs.

user_config.properties	
Keep.audit.for.x.days	Number of days to keep audit in the table.
notification.onSuccess.enabled	This property represents the flag set to enable notifications when a job runs successfully.
notification.onSuccess.emails	This property represents the e-mail address to send a message when a job has run successfully.
notification.onFailure.enabled	This property represents the flag set to enable notifications when a job fails.
notification.onFailure.emails	This property represents the e-mail address to send a message when a job fails.

### 2.3.2.3 Potential Failures/Errors

Database-related errors – This job will fail if the database is inaccessible.

### 2.3.3 Batch – Notification Purge

Launch Directory:	<install_dir>/batch/notification_purge
Launch Command:	ant run
Configuration Dir:	<install_dir>/batch/notification_purge/resources/ <install_dir>/config/common/
Configuration File(s):	notification_purge.properties user_config.properties

#### 2.3.3.1 Overview

This purge job deletes records from the NTF\_ATTACHMENT and NTF\_HISTORY tables; specifically, records are deleted that exceed (and, therefore, are older) than the designated number of days to keep notifications in these tables.

### 2.3.3.2 Configuration

notification_purge.properties	
job.name	Job name used in the job status notification messages.
purge.chunk.size	This property represents the number of records to be deleted at a time; size of delete chunk.
batch.success	ID of the batch job e-mail template. ("batch_success")
batch.failure	ID of the batch job e-mail template. ("batch_failure")

user_config.properties	
Keep.notification.for.x.days	Number of days to keep notifications in the table.
notification.onSuccess.enabled	This property represents the flag set to enable notifications when a job runs successfully.
notification.onSuccess.emails	This property represents the e-mail address to send a message when a job has run successfully.
notification.onFailure.enabled	This property represents the flag set to enable notifications when a job fails.
notification.onFailure.emails	This property represents the e-mail address to send a message when a job fails.

### 2.3.3.3 Potential Failures/Errors

Database related errors – This job will fail if the database is inaccessible.

## 2.4 Servers

The delayed queue (DQ) notification server is used for bulk notifications generated by batch jobs. This server may need to send more than 100,000 e-mails in a single day.

### 2.4.1 Server – Notification DQ Server

Control Directory:	<install dir>/server/notify_dq
Control Command:	ant start ant stop
Configuration Dir:	<install dir>/server/notify_dq/resources/
Configuration File(s):	NotificationServer.properties

#### 2.4.1.1 Overview

This server processes batch (bulk) e-mail, SMS, or other style notifications generated by batch jobs that generate thousands of notifications.

The model behind the *delayed queue* (or DQ notification server) is that the notifications it sends do not need to be real-time, and that notifications in response to individual user requests (such as registration) will use the *immediate queue* or IQ notification server. The DQ notification server does not intentionally delay notifications; it just handles requests for thousands at a time, which may be time-consuming.

If this server is stopped, notifications can still be queued; however, they will not be delivered. Once this server is back up, it will continue sending any notifications that have been queued.

#### 2.4.1.2 Potential Failures/Errors

If this server stops, it is typically because there was an interruption in the database connection. This interruption is often caused by a firewall that silently drops the TCP/IP connection without telling either end.

#### 2.4.2 Server – Notification IQ Server

The IQ notification server is used for one-off, time-sensitive notifications such as those associated with self-registration and immediate payment notifications.

Launch Directory:	<install dir>/server/notify_iq
Launch Command:	ant start ant stop
Configuration Dir:	<install dir>/server/notify_iq/resources/
Configuration File(s):	NotificationServer.properties

#### 2.4.2.1 Overview

This server processes time-critical e-mails, SMS, or other style notifications generated in response to user events (to include registration and profile changes).

The model behind the *immediate queue* (or IQ notification server) is that the notifications it sends must happen quickly (almost in real-time), and there are a limited number queued simultaneously. Separating them from batch-generated requests ensures that they do not get caught behind 100,000 bill notification e-mails.

If this server is stopped, notifications can still be queued; however, they will not be delivered. Once this server is back up, it will continue sending any notifications that have been queued.

#### 2.4.2.2 Errors

If the server stops, it is typically because there was an interruption in the database connection. This interruption is often caused by a firewall that silently drops the TCP/IP connection without notifying either end.