

PostalOne! Mail.dat Client Batch Processing

Can you upload Mail.dat[®] files to *PostalOne!*[®] consistently? Is your volume of work exceeding the number of hours in your day? Then it's time to introduce you to the unattended or batch processing capabilities of the Mail.dat Client, using examples that you can adjust to meet your needs.

We discuss the following topics. Click a topic to go directly there.

- Conventions
- What Is Batch Processing?
- What You Need
 - Example
- Get Your Metrics
 - Example
- Choose Your Method
- Work the Wizard
- Configure the Client
 - All Modes
 - One-Time Mode
 - Scenario 1
 - Scenario 2
 - Automation Batch Processing
 - Alternatives
 - Receipt Polling
 - Batch Processing Workflow
- Testing Testing
- Ready to Go
 - Example
- References

Versions of the *PostalOne!* Mail.dat Client and the Mail.dat Wizard used in preparing this document were available on December 11, 2009.

Conventions

The examples in this document use the following conventions:

- Files, folders, and command line entries appear in `this font`.
- Sample input data appears in *italics*.
- Buttons, links, and icons not labeled as such appear in **Bold**.



Technical Support Information

Mail Manager 2010

800) 624-5234

tech@bccsoftware.com

Mail Manager 2010 LE

866) 326-5168

techle@bccsoftware.com

Mail Manager FS

866) 326-0108

techfs@bccsoftware.com

Technical Document Questions/Comments:

techdocs@bccsoftware.com

What Is Batch Processing?

Batch processing is unattended processing. It automates *PostalOne!* file validation and upload, reducing the time spent on interactive work.

Do you have a time-consuming job to upload? Use the one-time option to send it overnight. Do you want to automate validation and upload? Set the daily start time, duration, break time, and folder assignments; then you're ready.

Think of it this way: Two workstations uploading Mail.dat jobs using interactive mode for eight hours a day is the rough equivalent to one workstation working in unattended mode for 16 hours a day after hours. The workstation running in unattended mode is available for other tasks during peak office hours. This illustrates the potential of batch processing to free resources for different use, saving both time and money.

It's a secure mode of processing, storing the encrypted user name and password with the Mail.dat Client installation, providing automated access to *PostalOne!* without a security certificate.

Although batch processing is available for both Windows and Solaris, this document focuses on Windows implementation because Mail Manager products are Windows-based.

What You Need

It's vital to establish a consistently high percentage of successful uploads BEFORE you implement batch processing on more than a small experimental basis using the Test Environment for Mailers (TEM). Why? Because a high failure or rejection rate (failure and rejection are different; we'll explain later) reduces your savings. Batch processing identifies the jobs that do not upload but it does not fix and resend them. That's up to you, which means there is time spent troubleshooting and resending jobs that you'd hope to devote to other things. Even resending jobs during the next unattended session could create backlog issues, so whether you choose to resend jobs interactively or in the next unattended session, the higher the failure or rejection rates, the less you save.

Example. In the "What Is Batch Processing?" section of this document, we cited the illustration of a workstation running unattended for 16 hours replacing two workstations running interactively for eight hours. What if jobs totalling four hours of upload time fail validation? Time must be spent on troubleshooting and correction, and the four hours of upload time must be accommodated. Whether this is done during the day or at night, that represents four hours of upload time that could have been put to alternative use. Up to four hours of new uploads may be delayed.

But since your *PostalOne!* uploads run like a fine-tuned machine, are you ready to begin? You need:

- *PostalOne!* user name and password
- Windows workstation with the following software:
 - Mail Manager 2010 with Mail.dat option OR Mail Manager FS
 - *PostalOne!* Mail.dat Client. Batch mode uses the following files:
 - `runBatchUpload.bat`: Validates and transfers Mail.dat jobs to *PostalOne!*
 - `runReceiptPoller.bat`: Obtains status reports from *PostalOne!* servers
 - `run-client_credentials.bat`: Changes the user name and/or password
 - `mdrsettings.conf`: Contains configuration and scheduling information. The examples in this document do not edit this file directly.
 - Calling program: The examples in this document use Windows Scheduled Tasks.
- High-speed Internet connection
- Method of recording job sizes and upload times

"What Is Batch Processing?" defined the motive for implementing batch processing. It is to achieve maximum efficiency in the use of resources that upload electronic documentation to *PostalOne!* Stating that motive in more quantifiable terms makes it easier to measure your success. For example, "I want to limit interactive use of the Mail.dat Client to one person on one workstation for one hour a day." Or, "I need to make the Mail.dat Client workstation available for Task ABC from 9 to 11 AM." Or, "I'd like to upload jobs taking two hours or more (jobs larger than xx megabytes) overnight."

But there are different approaches available; which is best for you? To answer that question, it is vital to quantify your current process. That is the objective of the next step.

Get Your Metrics

How much time do you spend uploading files to *PostalOne!*? How is your upload time distributed? When are the windows of opportunity for unattended uploads? Answering these and other questions is the objective of this step.

Table 1 on page 4 presents some issues to consider when choosing a batch processing strategy, and the related Mail.dat Client tools. Export reports from the Upload Metrics page to CSV files for further analysis.

More detailed information on the size, upload time, and outcome of each upload is available through the Validation/Upload History page. However, it is the most time-consuming to obtain, and cannot be exported.

Why are these metrics important? For a number of reasons, including these:

- **You will be more able to document the benefits of batch processing.** The more precisely you can describe your current process, the easier it is to find the benefits of change.
- **You are not alone.** You must schedule batch processing around other events that occur both during office hours and overnight. Office hour events include the time required to review and correct failed uploads or to run other interactive applications, while overnight events include such things as system backup and the *PostalOne!* maintenance cycle.
- **A NEW JOB MUST NOT BEGIN BEFORE THE OLD ONE ENDS.** To implement batch processing, use a program such as a scheduler to call batch files included with the Mail.dat Client, `runBatchUpdate.bat` for Mail.dat file sets and `runReceiptPoll.bat` for status receipts. These files can run simultaneously. However, to avoid data corruption, do not start a new upload session or a new receipt polling session while a previous one is still running. Knowing as much detail as possible about starting patterns and upload times will help you develop a schedule that minimizes this risk.

The following example illustrates the impact of design on the risk of process overruns.

Example. Your longest upload takes two hours; most others take 10 minutes or less. You average 43 jobs per day. Jobs are NOT made available at an even rate.

Good: Set `runBatchUpload.bat` to run from noon until 9 AM. `runBatchUpload.bat` checks for new uploads at noon and every 30 minutes thereafter. Any uploads discovered are processed before the next 30-minute wait begins. "Automation Batch Processing" on page 10 describes how to do this.

If you can always commit the workstation to the same amount of processing time, this is the best concept to follow, because it combines periodic checks for available work with fewer opportunities for overruns. You can use a scheduler to accommodate different starting times, but may have to reduce the time in which `runBatchUpload.bat` can run should the variance be too great.

Not as Good: Set `runBatchUpload.bat` to run several times between noon and 9 AM, with starting times based on workflow patterns. Each session begins with a check for new uploads and ends when their processing is complete.

This approach would be necessary if there were varying amounts of available processing time each day. It works as long as each session ends before the next one begins. But suppose that 30 jobs averaging 10 minutes processing time were available at noon, with the next session scheduled for 3 PM. The noon session would require five hours, two hours too long. Reduce this possibility by running fewer sessions per day, thereby reducing the number of checks for new uploads, or by matching the starting times more precisely to demand.

The requirements and related metrics in Table 1 on page 4 are intended as a starting point. Add whatever metrics and measurement methods you need to obtain a satisfactory description of your upload process. Should you be new to Mail.dat and *PostalOne!*, or if you anticipate growth not reflected in current metrics, you will have to incorporate projections as well.

Metrics in hand, it is time to determine a batch processing method.

NOTE



There is a way to see if the batch file has completed. The `runBatchUpload.bat` file and the `runReceiptPoller.bat` files use the Pause command as their last line. If you see the "Press any key to continue ..." prompt in a command window, the batch file will end once you comply. Otherwise, it is still running; end the `javaw.exe` process to close the window. Comment out the pause commands to run consecutive sessions of these batch files after hours.

Requirement	Metric	Source
<ul style="list-style-type: none"> Evaluate readiness to use batch processing in production. 	<ul style="list-style-type: none"> Success Rate 	<ul style="list-style-type: none"> Mail.dat Client>Upload Metrics>System Success Rate
<ul style="list-style-type: none"> Allocate sufficient unattended upload time. 	<ul style="list-style-type: none"> Processing Time History 	<ul style="list-style-type: none"> Mail.dat Client>Upload Metrics >Average Submission Time Export this report to Excel and extrapolate total upload times. This report does not display upload times well for small uploads. Obtain upload times per upload in the Mail.dat Client Validation/ Upload History page. You cannot export this data.
<ul style="list-style-type: none"> Accommodate tasks such as system backup and the updating and resubmitting of jobs that did not upload unattended. Start and end each day's processing at appropriate times. Run unattended processing on the appropriate days. 	<ul style="list-style-type: none"> Local and <i>PostalOne!</i> network maintenance schedules Personnel availability Usage requirements for <i>PostalOne!</i> workstation 	<ul style="list-style-type: none"> Local IT and management <i>PostalOne!</i>® Help Desk DMM® Advisory RIBBS®

Table 1: Metrics for Batch Processing planning

Choose Your Method

Now apply the metrics you have obtained to the choice of an unattended upload method. You should be able to answer the following questions:

- Will I change my current process?
 - Yes
 - No

If you answered "Yes," continue with the other questions.

- What changes will I make?
 - Upload time-consuming jobs after hours
 - "All Modes" on page 7
 - "One-Time Mode, Scenario 1" on page 8
 - Automate my upload process completely
- If automating completely, which line best describes my situation?
 - After hours, I will upload all files accumulated that day.
 - "All Modes" on page 7
 - "One-Time Mode, Scenario 2" on page 9
 - I will upload files as they become available.
 - "All Modes" on page 7
 - "Automation Batch Processing" on page 10
 - I'm still not sure because of scheduling considerations.
- How much time is available each day?
 - I have the same amount of uninterrupted processing time available each day.
 - "All Modes" on page 7
 - "One-Time Mode, Scenario 2" on page 9 if processing all files after hours
OR
"Automation Batch Processing" on page 10 if processing files throughout the day
 - I have varied amounts of uninterrupted processing time available each day.

5. If uninterrupted processing time varies, do the available times have a usable common factor?
 - Yes, I will use the factor as my Run Time and configure Automation Batch Processing Mode.
 1. "All Modes" on page 7
 2. "Automation Batch Processing, Alternatives" on page 11
 - No, I will configure Automation Batch Processing, using the smallest available time as my Run Time.
 1. "All Modes" on page 7
 2. "Automation Batch Processing, Alternatives" on page 11
 - No, I will use multiple sessions of `runBatchUpload.bat` without assigning a Run Time or Sleep Time in the Batch Configuration tab of the Mail.dat Client Setup dialog.
 1. "All Modes" on page 7
 2. "One-Time Mode, Scenario 2" on page 9

Numbered steps subordinate to bullets mean enough information is available to recommend a batch processing method. They don't mean much to you yet; let's address that.

The Client supports the following modes of operation:

- **One-time validation and upload:** Specify a location and start time. This is useful for processing long jobs at off-peak hours.
- **Automation Batch Processing:** Monitors a "hot" or "watch" folder for available work at designated intervals and processes what it finds. Controlled beginning and end over longer run times results in more frequent checks with less likelihood of session overruns.

Basic configuration is done almost exclusively through the graphical user interfaces of the Mail.dat Client and Windows Scheduled Tasks. The exception is One-time validation, which requires adding filenames to a batch file.

NOTE



Another command line requirement is to use the `run-client_credentials_tool.bat` file to change the user name and password in the `mdrsettings.conf` configuration file when doing so outside of the Client. You can edit the rest of this file with a text editor, but the user name and password fields are encrypted.

Now you might be wondering, "Does any of this affect how I run the Mail.dat Wizard?" The next section answers that question.

Work the Wizard

In a nutshell, batch processing works like this: You send your data to a designated "hot" or "watch" folder. A utility monitors that folder. If files are there, the utility processes them, then relocates them based on the processing's outcome.

The role of the Mail.dat Wizard is to generate Mail.dat files and place them in the appropriate folder for processing. Our examples use `C:\Postal1\Jobs\` as the "hot" folder or Jobs Repository. It must be the folder stipulated in the Mail.dat Client Settings window, Batch Configuration tab, Jobs Repository field. See Figure 3 on page 11.

The following settings in the **GENERAL** and **DATABASE** pages of the Mail.dat Wizard support batch processing:

- **GENERAL/Mail.dat/Location:** Select the folder from the Mail.dat Client Settings window/Batch Configuration tab/Jobs Repository field to save generated file sets there automatically (Example: `C:\Postal1\Jobs\`).
- **DATABASE: Do not check *Save Mail.dat database in unique folder*.** The subfolder structure is unnecessary; the Mail.dat files leave the Jobs Repository after processing.
- **DATABASE: Check *Zip Mail.dat database*** to conserve disk space and automate the order of file processing during validation and upload.
- **DATABASE: Check *Delete Mail.dat database after zipping*** to avoid Jobs Repository clutter.

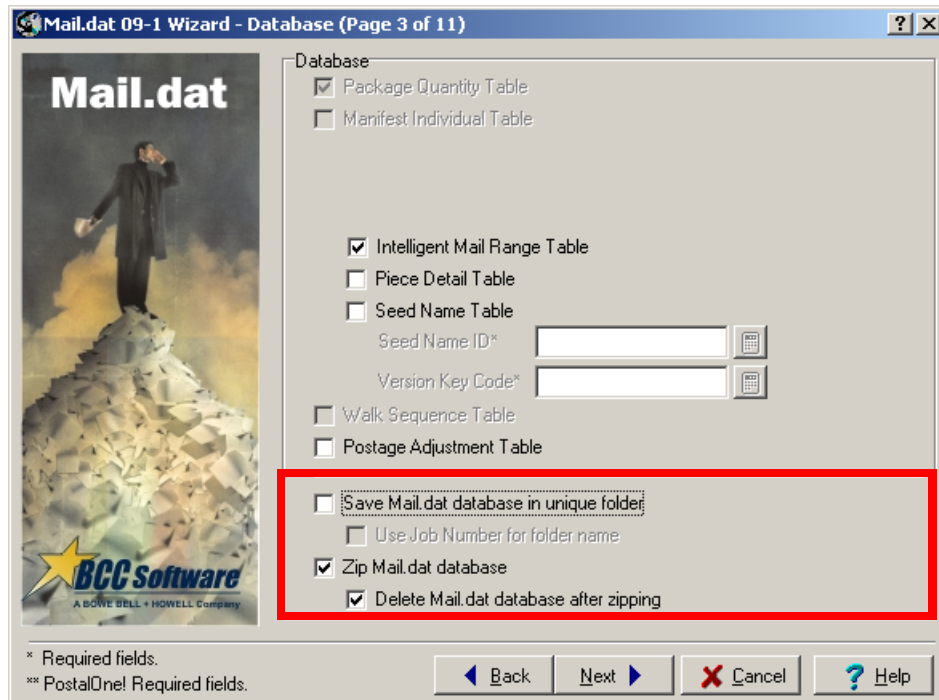


Figure 1: Database page settings that support batch processing

Configure the Client

You've done your homework and know what you want. The Mail.dat Wizard is assigning files to the Jobs Repository. It's time to set up the Mail.dat Client. Follow these steps:

1. Follow the steps illustrated in "All Modes" on page 7 to set up connectivity, user information, and input/output folders.
2. Do one of the following,
 - Follow the steps illustrated in "One-Time Mode, Scenario 1" on page 8 to upload a designated job or jobs on a designated day and time.
 - Follow the steps illustrated in "One-Time Mode, Scenario 2" on page 9 to implement batch processing where it is sufficient to check the Jobs Repository at the end of each day and process jobs overnight.
 - Follow the steps illustrated in "Automation Batch Processing" on page 10 to implement batch processing where multiple Jobs Repository checks per day are necessary and consistent amounts of time per processing period ARE available.
3. Follow the steps illustrated in "Receipt Polling" on page 12 to automate the receipt of status information from the *PostalOne!*[®] servers.
4. Follow the steps illustrated in "Batch Processing Workflow" on page 14 to take the configuration for a spin.

All Modes

All modes require you to assign folders for logging, input, output, and receipt polling. A log folder (see step 5) is required for both interactive and batch processing.

1. Log in to the Mail.dat Client.
2. Click **Settings**.
3. Click the General tab.
4. If required, enter a Proxy Server Address and Port Number.
5. Click **Browse** to select a Log File Location such as C:\Postal1\Logs. This folder stores logs generated for Client activities, and is required for interactive mode as well.

NOTE



The path names displayed in Settings follow the Solaris/Unix convention of separating folder levels with a forward slash ('/'), where Windows path names use a backslash ('\').

6. Select your Time Zone from the drop-down box.
7. Click the Batch Configuration tab.
8. Do not change the Username or Password unless the new account can use the USPS® Business Customer Gateway and Electronic Documents Exchange.
9. Assign a "placeholder" Run Time of 2 minutes and Sleep Time of 1 minute.
 - Always express these times in minutes.
 - These values allow "one-time" jobs to process assigned or available files and exit after a brief delay.
 - Run Time: Length of time each day that the watch folder is monitored.
 - Sleep Time: Interval between watch folder checks
10. Use the associated Browse button to designate a location for each of the following folders:
 - Jobs Repository: Input folder (C:\Postal1\Jobs)
 - Rejected Jobs: Jobs that terminate prior to validation. If submitted interactively, a FATAL ERROR message would appear. (C:\Postal1\Rejected)
 - Failed Jobs: Jobs that fail validation (C:\Postal1\Failed)
 - Successful Jobs: Jobs that pass local validation and attempt the upload to *PostalOne!* (C:\Postal1\Passed) They are subject to validation by the *PostalOne!* servers and may still not upload successfully.
11. Click **OK** to save your entries.
12. The next steps depend on the form of batch processing used:
 - See "One-Time Mode, Scenario 1" on page 8 to set up a job or jobs for one-time unattended processing.
 - See "One-Time Mode, Scenario 2" on page 9 to set up unattended processing for jobs saved to the Jobs Repository, when a consistent Run Time is NOT available.
 - See "Automation Batch Processing" on page 10 to set up automated processing for jobs saved to the Jobs Repository, when a consistent Run Time IS available.



Figure 2: Log File and Time Zone settings

One-Time Mode

These settings are in addition to those made in “All Modes” on page 7.

In these scenarios, the Mail.dat Client resides in the folder `C:\Postal1`. The Mail.dat Client is configured to look for jobs in the `C:\Postal1\Jobs` folder.

Scenario 1. You used the Mail.dat Wizard to generate the jobs `ABCD0001.zip` and `ZYXW0002.zip`. (These were for two different customers, hence two different ULCs.) The Wizard saved them to the `C:\Postal1\Jobs` folder. They are large, time-consuming jobs to upload and you would rather do so outside of peak office hours, starting at 6 PM.

First, modify the batch processing file to include the `ABCD0001.zip` and `ZYXW0002.zip` files:

1. Using a text editor, open `runBatchUpload.bat` from the `C:\Postal1` folder.
2. Edit the next to last line of `runBatchUpload.bat` to process the specific jobs. For this example:

```
%JAVA_HOME%\bin\javaw.exe %MEMALLOC% -client -classpath %CLASSPATH%
gov.usps.mdr.client.controller.CmdController uploadMode ABCD0001.zip
ZYXW0002.zip
```
3. Save the modified batch file with a new name, such as `runBatchUpload_once.bat`.
4. [Optional] Log in to the Mail.dat Client and validate these jobs so you can address issues that would prevent the scheduled upload. Log out, then exit the Client when finished.

Now, use Scheduled Tasks to set up the off-peak upload:

5. Access the Windows Control Panel. (The exact sequence will vary depending on both your version of Windows and your desktop settings.)
6. Double-click **Scheduled Tasks**.
7. Double-click **Add a Task**. The Scheduled Task Wizard opens.
8. Click **Next**. An application selection page opens.
9. Click **Browse**.

10. Browse to the C:\Postal1 folder.
11. Select `runBatchUpload_once.bat`.
12. Click **Open**. A new page appears.
 - a) Modify the task title as desired. (It defaults to the file name less extension, in this case `runBatchUpload_once`.)
 - b) Select *One time only* as the frequency of operation.
 - c) Click **Next**.
13. Assign a starting day and time.
 - a) Enter *6 PM* as the starting time.
 - b) Accept today's date as the start date.
 - c) Click **Next**.
14. Enter the Windows user name and password under which this task will run.
15. Click **Next**.
16. Click **Finish** to end the wizard.

Scenario 2. Now apply Scenario 1 to automating all jobs sent to the Jobs Repository, using the requirements from the Example on page 3 (43 jobs a day, 21 hours available, start time noon, longest job needs two hours).

First, verify that the batch processing file DOES NOT include specific files:

1. Using a text editor, open `runBatchUpload.bat` from the C:\Postal1 folder.
2. Delete any `Mail.dat`, `.hdr` or `.zip` file set names from the next to last line of `runBatchUpload.bat` so that it reads like this:

```
%JAVA_HOME%\bin\javaw.exe %MEMALLOC% -client -classpath %CLASSPATH%
gov.usps.mdr.client.controller.CmdController uploadMode
```
3. If you deleted file names in step 2, save the modified batch file.

Now, use Scheduled Tasks to set up unattended uploads:

4. Access the Windows Control Panel. (The exact sequence will vary depending on both your version of Windows and your desktop settings.)
5. Double-click **Scheduled Tasks**.
6. Double-click **Add a Task**. The Scheduled Task Wizard opens.
7. Click **Next**. An application selection page opens.
8. Click **Browse**.
9. Browse to the C:\Postal1 folder.
10. Select `runBatchUpload.bat`.
11. Click **Open**. A new page appears.
 - a) Change the task name to `runBatchUpload_1`.
 - b) Select *Daily* as the frequency.
 - c) Click **Next**.
12. Assign a starting day and time.
 - a) Enter a starting time of *12 PM*.
 - b) Select *Weekdays* to exclude the *PostalOne!*[®] maintenance window.
 - c) Accept today's date as the start date.
 - d) Click **Next**.
13. Enter the Windows user name and password under which this task will run.
14. Click **Next**.
15. Click **Finish** to end the wizard.

This is how Scenarios 1 and 2 compare:

- Scenario 1: At *6 PM* on the assigned date, the Scheduled Task `runBatchUpload_once` opens `runUploadBatch_once.bat`, which processes the assigned files, moving each to the appropriate output folder. Upon completion, `runUploadBatch_once.bat` exits, and `runBatchUpload_once` never runs again.
- Scenario 2: Each *weekday* at *noon*, the Scheduled Task `runBatchUpload_1` opens `runUploadBatch.bat`, which checks the Jobs Repository, processes whatever it finds, and moves each file to

the appropriate output folder. Upon completion, `runUploadBatch.bat` exits, and `runBatchUpload_1` runs the next *weekday at noon*.

What if a daily check is insufficient? Set up Scheduled Tasks like Scenario 2 to run more than once each day, but not frequently enough to risk conflicts. For example, to achieve seven Jobs Repository checks a day over 21 hours, your first attempt might set up tasks to run every three hours; at noon, 3 PM, 6 PM, 9 PM, midnight, 3 AM, and 6 AM. As usage patterns emerge, adjust the schedule accordingly. **Remember to comment out the pause statement that ends the batch file.**

This is an alternative to Automated Processing should different days have different windows of opportunity for batch processing. (This is how the batch processor that supports Mail.dat 08-2 uploads works.)

NOTE



To see other configuration settings, either check the advanced options checkbox before clicking Finish or right-click a Scheduled Task and click Properties. The task's Properties page opens. Its Settings tab includes options concerning task completion and task relation to power management policies. Considering disabling the Stop the task if it runs for ... option. The USPS® recommends that to end a session, first click the window in which the process is running and type Ctrl-C. If that fails, open Task Manager and select the Processes tab. Select javaw.exe, click End Process, and click Yes to confirm.

If the task does not run unattended and all other settings are correct, power management settings may be the cause. Try enabling the Wake the computer to run this task option.

Automation Batch Processing

If your batch processing goal is to process all files generated in a day after hours beginning at a certain time, then Scenario 2, adjusted to start at day's end, is a good fit. End of day comes, the Scheduled Task opens, the batch file checks the Jobs Repository, then processes the files. No other checks are needed as no one is around to generate work.

Require Jobs Repository checks more often, perhaps during the day to approximate real-time processing, and Scenario 2 reveals its drawbacks. Configure our example as one Scheduled Task and only the jobs available at noon upload that day. Using multiple Scheduled Tasks increases the frequency of Jobs Repository checks. but requires precise scheduling to avoid data corruption, should one upload session start before the previous one ends. Is there another way?

Suppose you could schedule unattended processing in a way that uses one Scheduled Task, checks the Job Repository throughout the day, and accommodates variable workflow? Enter the Mail.dat Client Automatic Batch Processing mode. Now setting up ONE Scheduled Task accomplishes what required many before. It's easier to prevent tasks from overrunning each other as well.

These settings are in addition to those made in "All Modes" on page 7. Our example continues to be seven jobs that run from noon to 9 AM.

1. Log in to the Mail.dat Client.
2. Click **Settings**.
3. Click the Batch Configuration tab.
4. Assign a Run Time of 1260 minutes (21 hours * 60 minutes/hour) and Sleep Time of 30 minutes.
 - Always express these times in minutes.
 - Run Time: Length of time each day that the watch folder is monitored.
 - Sleep Time: Interval between watch folder checks
5. Click **OK**.
6. Log out of the Client.

Now set up the Scheduled Task.

7. Access the Windows Control Panel. (The exact sequence will vary depending on your version of Windows and your desktop settings.)
8. Double-click **Scheduled Tasks**.
9. Double-click **Add a Task**. The Scheduled Task Wizard opens.
10. Click **Next**. An application selection page opens.
11. Click **Browse**.
12. Browse to the C:\Postal1 folder.

13. Select `runBatchUpload.bat`. Click **Open**. A new page appears.
 - a) Change the task name to `runBatchUpload_auto`.
 - b) Select *Daily* as the frequency.
 - c) Click **Next**.
14. Assign a starting day and time.
 - a) Enter a starting time of *12 PM*.
 - b) Select *Weekdays* to exclude the *PostalOne!* maintenance window.
 - c) Accept today's date as the start date.
 - d) Click **Next**.
15. Enter the Windows user name and password under which this task will run.
16. Click **Next**.
17. Click **Finish** to end the wizard.

Each *weekday at noon*, the Scheduled Task `runBatchUpload_auto` opens `runUploadBatch.bat`, which checks the Jobs Repository, processes whatever it finds, and moves each file to the appropriate output folder. Upon completion, the system "sleeps" for *30 minutes*, then checks the Job Repository again. This alternate cycle of checking, processing, and "sleeping" continues until 9 AM the next day, or until the last queued job is complete. `runUploadBatch.bat` exits; `runBatchUpload_auto` runs the next *weekday at noon*.

In our example, this method reduces the daily opportunities for process overruns from seven, (one at the end of each Scheduled Task) to one (should processing of the last queue of jobs run past noon).

The screenshot shows the 'Settings' dialog box with the 'Batch Configuration' tab selected. The 'Upload' section is highlighted with a red border. It contains two input fields: 'Run Time' set to 1260 minutes and 'Sleep Time' set to 30 minutes. Below this is the 'Batch File Locations' section, which includes four rows of text boxes and 'Browse...' buttons: 'Jobs Repository' (C:/Postal1/Jobs), 'Rejected Files' (C:/Postal1/Rejected), 'Failed Jobs' (C:/Postal1/Failed), and 'Successful Jobs' (C:/Postal1/Passed). The 'Username/Password' section is visible above the Upload section, with 'Username' set to 'JQMAILER' and 'Password' masked with asterisks. The 'Cancel' and 'OK' buttons are at the bottom right.

Figure 3: Batch Configuration tab with Automated Batch Processing

Alternatives. Automation Batch Processing is not always viable. What if your system is available for 21 hours on Monday and Thursday, 16 hours on Tuesday and Thursday, and not at all on Friday? Consider the following possibilities:

- **Less than five-day availability, same amount of time each day.** (Monday, Tuesday, and Thursday for 21 hours each day): Retain the Run Time and Sleep Time in Figure 3; set up one Scheduled Task for each available day.
- **Differing availability, usable common factor:** (7 hours on Monday and Friday; 14 hours on Tuesday and Thursday, 21 hours on Wednesday) If the greatest common factor (7 hours, close to overnight) of the available processing times is of sufficient length, use it as the Run Time, in this case seven hours or 420 minutes. Set up Scheduled Tasks as follows:
 - If one starting time is common to all days, one Scheduled Task may be able to cover seven hours' processing time per day. In this example, Tuesday and Thursday would require an additional Scheduled Task each; Wednesday would require two.
 - If there are no common starting times, each day requires its own Scheduled Tasks (one each for Monday and Friday, two for Tuesday and Thursday, three for Wednesday).
- **Differing hours of availability, no usable common factor:** What if your available run times were 17 hours, 19 hours, and 23 hours? Each of these is a prime number, with factors of itself and one. Convert these times, to minutes, and the greatest common factor is 60.
 - 23 hours = 1,380 minutes = $23 \times 2 \times 2 \times 3 \times 5 = 23 \times 60$
 - 19 hours = 1,140 minutes = $19 \times 2 \times 2 \times 3 \times 5 = 19 \times 60$
 - 17 hours = 1,020 minutes = $17 \times 2 \times 2 \times 3 \times 5 = 17 \times 60$

How could you use Automation Batch Processing to schedule all available hours?

1. Set your Run Time to 60. (Sleep Time remains 30.)
2. Start a new Scheduled Task every available hour. Use weekly Scheduled Tasks for common starting times.

Is this approach consistent with the goals of Automation Batch Processing?

- You DO achieve multiple Jobs Repository checks based on the Sleep Time.
- You DO NOT reduce the possibility of process overruns; you need over 20 Scheduled Tasks.

Automation Batch Processing does not simplify your life this way. A compromise is necessary. You have the following options:

- If it is sufficient to meet demand, set the Run Time to the smallest alternative (in this case 17 hours or 1,020 minutes).
- OR
- Replace the Run Time and Sleep Time with extremely small values, such as a Run Time of 2 minutes and a Sleep Time of 1 minute, and use "One-Time Mode, Scenario 2" on page 9. There will be one Jobs Repository check at the beginning of each Scheduled Task.

NOTE



Depending on available starting times, either compromise may require similar numbers of Scheduled Tasks. However, the common Run Time setting should result in more Jobs Repository checks and offers more control over process duration. Users of the Scenario 2 option will have more hands-on responsibility for managing the input of jobs into the Jobs Repository, as the only ways this option ends is either by user intervention or after it has processed all discovered jobs, regardless of time allotted on an operations plan.

Remember to comment out the Pause command in your batch file if it is scheduled to run more than once after the office closes.

Receipt Polling

There are multiple levels of processing as a job navigates *PostalOne!*[®] Can we monitor an upload's status without going online? With status receipts, the answer is a qualified Yes. Status receipt files document an upload's milestones such as being loaded into the database and postage statement generation. Eventually they will reflect USPS[®] actions such as MPU Edits and Postage Statement Reversal. While configuration and scheduling resemble that of batch processing, receipt polling introduces additional issues of file format and naming. Receipts can be text or XML files, and naming options differentiate receipts by such things as Job ID and upload date.

Receipt polling can run in parallel with batch processing. Like batch processing, one receipt polling session must not begin unless the active one ends. UNLIKE batch processing, you cannot edit

`runReceiptPoller.bat` to obtain receipts for specific jobs, such as in the batch processing example from “One-Time Mode” on page 8.

Here’s a way to configure receipt polling for the sample job running weekdays from noon to 9 AM. Let’s run receipt polling in parallel.

1. Log in to the Mail.dat Client.
2. Click **Settings**.
3. Click the Receipt Files tab.
4. Click **Browse**. Designate `C:\Postal1\Receipts` as the Receipts Repository.
5. Select `ASCII` as the Receipt Format. Retain the default Delimiter.

NOTE

This is arbitrary. The format you assign depends on the application using the files.

6. Delete the default Receipt File Name of `StatusReceipt` to prevent receipt file overwrites. When Receipt File Name is blank, the default naming convention combines the Mail.dat files’ Job ID, Historical Sequence number, and upload date/time, followed by the Receipt Format file extension. The Receipt Polling Configuration/Configuration and Setup section of the *Postal Service Mail.dat Technical Guide* (section 5.2.5.1) includes other file naming options.
7. Assign a Receipt Poll Run Time of 1260 minutes (21 hours * 60 minutes/hour) and Sleep Time of 30 minutes.
 - Always express these times in minutes.
 - Run Time: Length of time each day that the watch folder is monitored.
 - Sleep Time: Interval between watch folder checks
8. Click **OK**.
9. Log out of the Client.

Now set up the Scheduled Task.

10. Access the Windows Control Panel. (The exact sequence varies, depending on both your version of Windows and your desktop settings.)
11. Double-click **Scheduled Tasks**.
12. Double-click **Add a Task**. The Scheduled Task Wizard opens.
13. Click **Next**. An application selection page opens.
14. Click **Browse**.
15. Browse to the `C:\Postal1` folder.
16. Select `runReceiptPoller.bat`. Click **Open**. A new page appears.
 - a) Change the task name to `runReceiptPoller_auto`.
 - b) Select *Daily* as the frequency.
 - c) Click **Next**.
17. Assign a starting day and time.
 - a) Enter a starting time of *12 PM*.
 - b) Select *Weekdays* to exclude the *PostalOne!* maintenance window.
 - c) Accept today’s date as the start date.
 - d) Click **Next**.
18. Enter the Windows user name and password under which this task will run.
19. Click **Next**.
20. Click **Finish** to end the wizard.

Each *weekday at noon*, the Scheduled Task `runReceiptPoller_auto` opens `runReceiptPoller.bat`, which checks the *PostalOne!*[®] servers for new status receipts and downloads whatever it finds to the Receipts Repository. Upon completion, the system “sleeps” for *30 minutes*, then checks *PostalOne!* again. This alternate cycle of checking, downloading, and “sleeping” continues until 9 AM the next day, or until the last queued receipt is downloaded, at which time `runReceiptPoller.bat` exits; `runReceiptPoller_auto` runs the next *weekday at noon*.

Batch Processing Workflow

Configuration complete, it's time to try a dry run. Regardless of mode, here are the steps:

1. Generate the Mail.dat files, sending them to the Jobs Repository. This must be finished before the Scheduled Task begins if you don't use Automatic Batch Processing.

NOTE



You may want to log the files sent to the Jobs Repository, the processing outcome, and follow-up measures. This helps in quantifying your savings and tracking customer work.

The screenshot shows the 'Settings' dialog box with the 'Receipt Files' tab selected. The 'Receipts Repository' field contains 'C:/Postal1/Receipts' and has a 'Browse...' button. The 'Receipt Format' section has radio buttons for XML, ASCII (selected), and Other, with a 'Delimiter' section containing radio buttons for Space (selected), Tab, and Other. The 'Receipt File Name' field contains 'StatusReceipt'. The 'Receipt Poll Run Time' is set to 1260 minutes and the 'Sleep Time' is set to 30 minutes. There are 'Cancel' and 'OK' buttons at the bottom right.

Figure 4: Receipt Polling settings

2. Log out of the Client before any Scheduled Tasks begin.
3. Once the Scheduled Task completes, check the outcome by inspecting the contents of the output folders. (Check for the "Press any key to continue ..." prompt if you didn't comment out the Pause command. The check need not be immediate, that's the "unattended" part.)
 - Files in the "Passed" folder uploaded. Use Status Receipts to track their progress through *PostalOne!*
 - Files in the "Rejected" folder never made it to full-fledged validation. The Mail.dat Client validation.log file may hold clues. Often It's a matter of data entry. Are you sure, for example, that you entered the correct CRIDs? Fix the errors in the Mail.dat Wizard and try again.
 - Files in the "Failed" folder went through validation and did not upload, or failed the "Just Validate" option. The Validation/Upload History page should offer an explanation. Often it involves issues of status updates, such as attempting to upload containers marked "Ready to Pay" twice. Some of these issues can be corrected in the Mail.dat Wizard, others require directly editing the Mail.dat files or re-creating the presort.
4. Check the Receipts Repository for updates on jobs that uploaded. Validation procedures within *PostalOne!* may raise new issues.
5. Upload revisions either interactively or during the next unattended session.

The fewer "Failed" and "Rejected" files you find, the greater your savings will be.

WARNING *The Client does not move files that you process interactively. Move them to an appropriate result folder yourself.*



Testing Testing

So everything's set up, does that mean that it's time to go live?

Maybe, maybe not. Are you certain that you've met the objectives broadly defined in "What Is Batch Processing?" on page 2 and quantified in "What You Need" on page 2? (You did quantify your goals, right?)

If you've used batch processing to submit test files to the USPS® and passed, you know that your data is good and you made batch processing work. If you passed in interactive mode, you know that your data is good, but you have yet to try unattended uploads. In either instance, the ultimate challenge is scheduling. Your schedule must protect your data by starting and stopping the batch processing files at appropriate times. You'll want to verify that your unattended upload schedule meets your goals and preserves your data before implementing it in production. The TEM is available for that purpose.

You'll be ready to go live when you can answer Yes to the following questions:

- Does the Mail.dat Wizard save my files to the correct input folder?
- Does unattended processing move each file to the correct output folder?
- Do I re-process unsuccessful uploads in a manner consistent with the folder assignments in the Batch Configuration tab?
- Do Scheduled Tasks begin on time and run on designated days?
- Does the amount of time allocated to each Scheduled Task meet demand?
- Do the assigned starting times for each Scheduled Task enable one Task to end before the next one starts?
- Is the time allocated for Status Receipt polling adequate for the number of uploads?
- Is the Status Receipt file format appropriate?
- Is there a process in place for logging each job and its milestones as it makes its way through the *PostalOne!* system?
- Does batch processing produce a net savings in time and resources after re-processing jobs that do not upload?
- Do the net savings in time and resources meet the goals I defined?

Here are a few suggestions for testing:

1. Configure the Mail.dat Wizard for each test job as discussed in "Work the Wizard" on page 5.
2. Use the Validate Only option in the Mail.dat Client to predict the outcome of batch processing before invoking a Scheduled Task. Leave the jobs that fail validation alone; part of the testing is to verify the disposition of jobs to the appropriate output folders. (Be sure to log off the Client before any Scheduled Tasks begin.)
3. Run the Success Rate report weekly to gauge improvements in (or maintenance of) quality.
4. Test during the day at first, for short periods. For example:
 - a. Run a single job similar to "One-Time Mode, Scenario 1" on page 8 at a designated time such as 1 PM.
 - b. If using One-Time Mode to automate general processing, first test the ability to process the Jobs Repository content at the assigned days and times. Monitor which input files are processed at which times, and if overruns result. Do not go live until you can process jobs in a way that avoids overruns. This may mean changing the times that you run a job through the Mail.dat Wizard or storing Wizard output to a temporary folder and moving it to the Jobs Repository manually or with a script.
 - c. If using Automation Batch Processing Mode, save several small Mail.dat file sets to the Jobs Repository. In the Mail.dat Client, set the Run Time to 60 minutes and the Sleep Time to 10 minutes. Set up a one-time Scheduled Task to start in 10 minutes. When it begins, generate more Mail.dat file sets to the Jobs Repository. Do they process?
 - d. If using Automation Batch Processing Mode, run the previous task, but see the effects of having new work available at the last Jobs Repository check. Do the available jobs process to completion, beyond the stated Run Time if required?

- e. If using Automation Batch Processing Mode, increase the Run Time and Sleep Time until they reach production values.
5. Develop a procedure for processing jobs that do not upload consistent with the output folder assignments in the Batch Configuration tab. Rejected Files jobs do not appear on the Validation/Upload History page; consult the `validation.log` file. Or re-validate them interactively to reproduce the issue, then correct and resubmit.

Ready to Go

Combine consistently error-free file sets with appropriate configuration and scheduling and you are on the way to more efficient use of the time and resources used with *PostalOne!*[®] and the Mail.dat Client. But periodic re-evaluation is necessary both to monitor compliance with your original goals as well as to accommodate change such as growth, software enhancement, and new postal requirements. There could be changes in both the number and size of jobs, leading you to re-evaluate:

- Method of batch processing used
- Frequency and duration of Scheduled Tasks
- Type of Internet connection
- Hardware upgrades to storage, RAM and CPU
- Networking

Example. You've run Automation Batch Processing mode for a year and a half. When you began, one system could easily process your workload; you had excess capacity. As you grew, the jobs grew in number and complexity until the required processing time has exceeded the time available. To put this in numbers, the overwhelming majority of uploads require 10 minutes or less. At 10 minutes per upload, the number of jobs cited in previous examples, 43, would require just over seven hours of upload time. Suppose the average number of daily uploads increases to 91 and average processing time grows to 13 minutes, as jobs became larger and more complex. That translates to almost 20 hours a day for uploading alone, without considering the break before each Jobs Repository check. The machine that easily handled 43 jobs a day is now struggles to meet demand. Contractual obligations make upgrades to your workstation and Internet connection unlikely, but a second workstation is available.

Here is a solution for a single-user Mail Manager installation:

1. Install and configure the Mail.dat Client and Scheduled Tasks on the new workstation.
2. Share the Jobs Repository folder on the new machine to users of the Mail Manager workstation, giving them Read and Write permissions.
3. When running the Wizard, use the General page Location field to distribute jobs between Jobs Repositories. The Wizard recognizes UNC names in case you didn't map a drive letter to the second workstation's share.

Unattended processing runs independently on each workstation according to its Mail.dat Client settings and Scheduled Tasks.

Need two Mail Manager users? Try this:

1. Upgrade Mail Manager to a networked version.
2. Use either the Local or Remote Server installation. (Use Local if you only connect two workstations.)
3. Assign postal data files and job data folders to shared locations.
4. Follow steps 1 and 3 above for each Mail.dat Client installation.

Using the networked version to process Mail Manager jobs facilitates centralized management of *PostalOne!*[®] Job IDs and Intelligent Mail Full Service compliance in assigning serial numbers, with less chance of compromised performance. Running Scheduled Tasks at each workstation significantly increases available starting times, by how much depends on how each machine is configured.

Apply the process used to implement batch processing to its evolution, and you will have a useful tool for years to come.

References

See your Mail Manager product's User Manual and online Help for Mail.dat Wizard information. Obtain manuals in PDF format from the BCC Customer Portal: <https://portal.bccsoftware.com/support/techManuals.aspx>

See Chapter Five of the *Postal Service Mail.dat Technical Specification*, http://ribbs.usps.gov/intelligentmail_guides/documents/tech_guides/PostalOneMaildatTechSpec.pdf for information on the Mail.dat Client, including the Solaris version.

Visit IDEAlliance, <http://www.maildat.org>, for Mail.dat licensing information and detailed version specifications.

Send questions and comments concerning BCC technical documents to techdocs@bccsoftware.com.

Published: 12/21/2009

NOTES

**Technical Support Information**

Mail Manager 2010

800) 624-5234

tech@bccsoftware.com

Mail Manager 2010 LE

866) 326-5168

techle@bccsoftware.com

Mail Manager FS

866) 326-0108

techfs@bccsoftware.com

Technical Document Questions/Comments:

techdocs@bccsoftware.com