Windows XP Automated Maintenance by Fred Langa

The problem isn't in Defrag, which is indeed fully schedulable, but rather in the inconsistent way that Microsoft built front ends for XP's self-maintenance tools. For instance, XP includes a complete, automated "Wizard" for its Backup applet; the Wizard walks you through the process of setting up fully-automated backups via Task Scheduler. But there's no similar Wizard or built-in hooks to Task Scheduler for other tools, like Defrag.

This can create a kind of vicious circle of underuse for those latter tools. For example, Defrag can take hours to run, especially if it hasn't been run in a while. If the only obvious way to run it is to trigger it manually, many users won't bother because it's inconvenient and ties up the PC for lengthy periods. Over time, the PC's files become more and more fragmented--scattered around the hard drive, piecemeal, instead of being stored in neatly contiguous wholes--which leads to a loss in drive responsiveness, extra wear, extra noise, and heat; and which can make some file-recovery operations (such as undeletes) less reliable. Because the drive is getting more and more fragmented, running Defrag would now take even longer to run, so the users put it off even more--on and on.

In contrast, when Defrag is used regularly, it usually only takes minutes, not hours, to run to completion, restoring normal performance and reducing wear, noise, and file-recovery problems. But that first use of Defrag can actually take several hours, so you'd think that Microsoft would have made it easy to run Defrag as an automated task, say, in the middle of the night when the PC wasn't in use and when it wouldn't matter if it took a while to complete.

Microsoft didn't, but we will. Indeed, it's possible to fully automate Defrag and myriad other tasks in XP that don't have a built-in scheduler.

In the rest of this article, we'll start with the basics to make sure everyone, even those new to Windows, are on board. But we'll end with some advanced ideas that can let your PC perform multiple automated unattended maintenance tasks of arbitrary complexity, and in whatever sequence you choose--and all for free, using only the tools built into XP. For example, if you want to have your PC automatically wake itself up every night at 3 a.m., clean up your hard drive, backup all your files, defrag every disk or partition in your system, and then go back to sleep--no problem! In fact, it's easy.

Let me show you how simple it can be, using Defrag as the working example.

****

Automated, Unattended Defragmentation

The normal, manual way to run Defrag is to click to Start/Control Panel/Performance and then select "Rearrange items on your hard disk to make programs run faster." All those clicks, and all that verbiage, simply starts the Defrag.Exe program (usually located in the Windows/System32 folder). When you run Defrag this way, it comes up with a graphical front end that requires still more pointing and clicking to make anything actually happen. So, this approach is useless for automated defrags when no one will be there to point and click as needed.

You get the same results if you search for "Defrag" in XP's help file: The primary search results ("Using Disk Defragmenter" and "Disk Defragmenter") offer links to the graphical, manual version of Defrag.

But Defrag can run just fine with no graphical front end at all and that's the key to using it in an automated fashion. To use Defrag (and other, similar system tools) this way, you launch the tool via a command line plus any "switches" you want to use to modify the file's behavior. (If you're not familiar with these terms, don't worry:)

To see how simple this really is, open an empty XP "command window": Click to Start/Run and then type the word "command" (without the quotes) in the Run line, and click OK. A command window, usually a mostly black box, will open. There'll be a blinking prompt at the end of a line of text. The text indicates your location within your hard drive's folder structure; and the blinking prompt shows you where anything you type will be entered. You can type any valid XP command there and XP will perform the action. You also can launch programs by typing the name of the tool, and that's what we're interested in here.

Almost all command-line tools offer simple, built-in help that you access simply by adding " /?" or " -?" (without the quotes) to whatever command or tool you're trying to learn about. For example, we're interested in Defrag here, so type DEFRAG /? (note the blank space after DEFRAG) and hit enter. This calls up simple help about Defrag in the form of a list of available Defrag options, and a template showing you how to enter Defrag commands in the correct format and order.

Using the Defrag help text as a guide, you can see that manually triggering Defrag is simplicity itself: The command "DEFRAG C:" (without the quotes) would tell Defrag to work on your C: volume or drive, for example. "DEFRAG D:" would tell Defrag to work on the D: drive. "DEFRAG C: -a" would tell Defrag to analyze how much your C: drive is fragmented, but not to actually perform a defragmentation. And so on.

In fact, if you type DEFRAG C: now, and hit enter, Defrag will run. But don't do it yet, because we're looking to automate the process, not to run it manually.

Command-Line Automation

Using the above example, you've seen how "commands" are just simple lines of text. You can enter these manually, one at a time, but you also can use a special kind of plain-text file to record the commands once, for reuse many times.

For example, open a new, empty Notepad and enter one line of plain text: DEFRAG C:

Now, click to Notepad's File/Save As menu. Navigate to your Desktop in the "Save In" portion of the dialog. In the "Save As Type" scroll box, scroll down to the "All Files" type (instead of the default "Text Documents"). Finally, in the "File Name" area, name your new file "DEFRAG C.BAT" (or any similar, obvious name ending in ".BAT"). Then, click Save. The file should be added to your desktop with a .BAT extension (instead of a .TXT extension).

You've just created a simple "batch" file (that's what the .bat file extension means). It's a special purpose plain-text file that's used as a kind of script to send commands to the operating system. In this case, the file contains just one command, but you can add many commands--in fact, a whole "batch" of them (hence the name)--and they'll be processed one at a time, in the order you list them.

For example, if you have other drives or partitions, you can either defrag them via separate batch files (a "DEFRAG D.BAT" file could contain just the line "defrag d:" for example), or you can enter the lines serially into one batch file. I have a batch file called "DEFRAG_ALL.BAT" for example, and it contains these lines:

DEFRAG C:
DEFRAG D:
DEFRAG E:
DEFRAG F:

When run, "DEFRAG_ALL.BAT defrags C, then D, and so on, one after the other until the last command is processed.

You can trigger a batch file either by manually clicking on the file, or--this is the key to automating any command-line task you add to a batch file--you can use Task Scheduler to run the batch file whenever you want. Done this way, whatever commands you place in the batch file will run at the designated time, one after the other.

Using Task Scheduler To Complete The Process

Click to Start/Control Panel/Performance and Maintenance and then click on "Scheduled

Tasks." Once the Task Scheduler opens, click on "Add Scheduled Task" and a "Wizard" will open to guide you through the process. Click Next.

When asked to "Click The Program You Want Windows To Run..." select the Browse button, and navigate to the batch file you just created in the previous step. Once you've selected the batch file, a new dialog will open, asking for a name and rough schedule for the new task you're creating. Enter the name you want, and then select an appropriate rough schedule. For example, if you want your system to defrag every day, select "Daily." Click Next.

In the next dialog, you can refine the schedule a bit: Pick a start time, a start date, and a frequency. For example, if you want Defrag to run every night in the middle of the night, you'd pick a suitable "Start Time" (say, 3 a.m.), select the "Every Day" option, and then set the start date to be the very next night. Click Next.

Depending on how your system is set up, you may then be presented with a dialog asking for your password; this will allow the automated task to get past any logins that may be required to run the scheduled task. Just enter your normal password, as indicated, and click OK. (Note: Most system-maintenance tasks run best from an account with Administrator permissions.)

The Wizard will then tell you that your basic scheduling is done, but also will offer an "Advanced Properties" check-box option. Select this option, and click Finish.

When the Advanced Properties opens, click the Settings tab. If this is the only automated task you'll have running, many of the options on the Settings dialog may not matter. But if there's any chance of another task running at or near the same time as the new task you're creating, set the "Idle Time" options accordingly. I find I get good results from setting the task to start "only if the computer has been idle" for at least 10 minutes; and to "retry the task" for four hours, or 240 minutes. Although Task Scheduler isn't great about preventing tasks from competing with each other, these settings will help to prevent such contention. Note the "Power Management" options on the same dialog. For laptops, the "Don't Start if the computer is on batteries" and "Stop task if battery mode begins" may be useful to prevent your system from running itself down when you're not connected to a wall socket. But all systems--laptops and standard PCs--can benefit from the "Wake the computer to run this task." This setting will bring your PC out of sleep or standby mode, if necessary, to run the scheduled task.

Close out the open dialogs, and click Next.

If you do nothing further, the task should run at the next-scheduled run time. But if you want to test your new task to make sure everything's set up OK, you can manually trigger it now: Still in the Task Scheduler, right click on the task you just created and select Run.

If the task is a long one--and defrags can take a long time indeed, especially if not run very often--you may wish to abort the task once you're sure it's running OK. Then you can use your PC normally.

At the end of the day, let your PC's power-control system put the PC into sleep or standby mode (not fully off). At the appropriate time, the PC will wake up, and run the scheduled task to completion.

Revised 7/17/2006 JMM

Advanced Users: Extending The Process

Any command--any command at all--that you can enter manually can also be placed in a batch file and run from the Task Scheduler in the above fashion. But you also can gang many tasks into one batch file, which simplifies scheduling and also ensures that one task will finish before the next one starts. I find this latter feature the strongest argument in favor of using batch files for system maintenance. Yes, you can enter command-line arguments directly into Task Scheduler, but to prevent multiple tasks from overlapping or running simultaneously, you'd either have to depend on Task Scheduler's imperfect ability to detect whether or not another task has completed or use sheer guesswork in building your schedule to allow sufficient time for one task to end before another begins. A sequential batch file avoids these problems.

For an example of a more complex batch file, see "CleanXP.Bat" which uses some "hidden" features of XP's System Agent tool and the Disk Cleanup Wizard, along with a series of manual commands, to rid your hard drive of all manner of hard-to-remove Temp files, potentially hundreds of megabytes worth.

You can easily add the Defrag command (as explained earlier) to the end of the CleanXP batch file, letting your hard drive reorganize itself after each cleaning.

Or, you could go even further by using a single batch file to clean your drive, reorganize itself, and then back up your files.

This may not be obvious because XP's backup "Wizard" has its own built-in scheduling function; on its own, it won't integrate with other cleanup functions. But you can use the Backup Wizard to build the proper command string for an automated backup, and then lift and use that string in your own cleanup procedure.

Here's how: You first create a scheduled backup the normal way, using XP's Backup and Restore Wizard. Step through the complete process, including setting up all the time and date. It doesn't matter what schedule you choose, as long as you complete all the steps in the Wizard.

When you're done, open the Task Scheduler, find the backup task you just created, and right click on it, selecting Properties. A dialog box will then open. Copy the entire contents of the "Run" line in the Task tab. The Run line may be very, very long because it contains the complete command for triggering the backup you just created, including all options. Paste this line, exactly as-is, to the end of your cleanup/defrag batch file; it will run as the last step in the batch process, and perform a backup exactly as you set it up in the Backup and Restore Wizard. In fact, you can go back to Task Scheduler and delete or disable the Wizard-created backup, as it's now redundant.

[A note to Windows 2000 users: Alas, even though XP is built on Win2K technology, things are not always so simple: For example, to automate Backups in Win2K, you have to use a more complex and indirect command line to open the Management Console, and run the Defrag "snap-in" from there. See "Reader Defrag Tips And Info: Win2K" for more information.]

Limited Only By Your Imagination

XP can run any command-line program or standalone command this way, letting you build custom maintenance sequences that do exactly what you want, when you want, in the exact order you want. It's enormously powerful, surprisingly easy, and limited only by your ingenuity.

How do you automate your maintenance tasks? What tools do you use? How do you ensure that maintenance tasks don't overlap, and that each task has sole use of the PC while it runs? Are there better ways of scheduling tasks that don't normally integrate with the Task Scheduler? Join in the discussion!

Editor's Note: To read more columns by Fred Langa and check out InformationWeek.com's new Windows Tech Center.

Revised 6/1/2009 JMM