



Pairs

*By Ed Wilson, Microsoft Scripting Guy
Winter Scripting Games 2014*

Your company has decided to have secret pals to improve morale. The idea is that each person in the department will be assigned the name of a team mate, and once a quarter you have to hide a low cost gift on his / her desk. Your manager has appointed you to write the script that will create the random name assignments. Here are the names of the people in the department for which you need to create the pairs:

Syed, Kim, Sam, Hazem, Pilar, Terry, Amy, Greg, Pamela, Julie, David, Robert, Shai, Ann, Mason, Sharon

An appropriate output is simple two names: such as

Sam, Hazem
Ann, Sharon
Syed, Terry
<etc>

Optionally the pairs should be saved to a file. The files should be named such that it is easy to see when they were created.

There are 16 names in the list, so that would be 8 pairs BUT your solution must work with more pairs than that. In addition, it should issue a warning if there is an odd number of names. At that point, you should be given the option to select a person to have TWO secret pals.

One of the project managers in your company heard about this from your manager and decided a similar approach could solve one of his problems. He is running a development project and depending on the phase of the project could have anywhere from 8 to 50 people working on the project. The software being developed is very important to the company's future plans so he has the developers working in pairs so that knowledge is shared.

The project manager needs you to modify your code so that:

- It can handle varying numbers of people
- The results of each run need to be saved
- The pairings will be run every 1 to 2 weeks
- The PM wants to be able to specify up to 5 people (0-5) who will be the primary of any pair. They should never pair with another primary
- The primaries can change at each stage of the project
- No two people should work with each other until they have been paired with at least 4 other people
- The code should have the option to email the people to inform them who their next partner will be. The project manager should be included in any communications.

Your code will be used by other project managers with similar needs so should be production ready with:

- Ability to optionally report on progress
- Full error checking, reporting and handling
- Ability to accept pipeline input where appropriate
- Help is available
- Input is validated

You are expected to have two solutions – the one for your manager and the one for the project manager. You are encouraged to share code between the two solutions.

In your entry submission, include a transcript that shows you running the command as described in this scenario.



Finally! Public PowerShell Classes from Don Jones and Jason Helmick!

Get the schedule from
<http://events.concentratedtech.com>

PolicyPak

WHAT YOU SET IS WHAT THEY GET.

Got **Java**? Got **Firefox**? Got **Flash**?
Smackdown pop-ups and lock'em down.



MVP Jeremy Moskowitz demos **secret** toolkit

www.PolicyPak.com



SAPIEN POWERSHELL TOOLS



SAPIEN
PowerShell
STUDIO™



PrimalScript®



iPowerShell™ Pro

TO LEARN MORE GO TO www.sapien.com

PrimalScript, SAPIEN PowerShell Studio, SAPIEN Software Suite (S3), iPowerShell Pro, ChangeVue, PrimalSQL, and PrimalXML are trademarks of SAPIEN Technologies, Inc. All other logos, trademarks, and service marks are the property of their respective owners.
© 2002-2014 SAPIEN Technologies, Inc. All Rights Reserved.

Key Criteria

These are some of the main items our judges will consider. You do not need to meet all key criteria, but you may earn extra points for doing so. This list is intended as a summary, and does not override the specifications of the scenario above.

- Consider the practices in *The Community Book of PowerShell Practices* (linked at <http://powershell.org/wp/newsletter>)
- Avoid aliases, except for –Object cmdlets; avoid positional parameters and truncated parameter names.
- Use appropriate error handling.
- Use appropriate means of displaying output, progress messages, errors, etc.
- When appropriate, manage pipeline input correctly
- When appropriate, validate input via parameter validation attributes
- Provide help for all scripts and functions, including examples
- Script filenames should include production date for versioning
- Use modular programming practices to maximize opportunities to share code
- Pairs assigned randomly as specified
- No name used twice
- No person is their own secret pal
- Odd numbers managed appropriately
- Varying numbers handled correctly
- Primaries can be specified and changed
- No pairings repeated with specified timeframe
- Option for email

As is often the case in Windows PowerShell, there will be many ways to complete these objectives. In most cases, judges will prefer approaches that:

- Perform well under the load specified
- Leverage built-in functionality of Windows PowerShell rather than reinventing the wheel
- Are the most straightforward and easy to read and understand