Spawning Processes into Interactive Sessions

Starting a Windows process using .NET is normally an easy thing.
You call Process.Start() with a little setup, and the process runs.

But, that starts a process in the same Windows session as the thread that calls Process.Start().

This, of course, doesn't work for a Windows Service, since a service runs in a non-interactive Windows session (session 0).

For a Windows Session to spawn processes into interactive user sessions, a few steps are required.

Specifically, we have to:

- Identify the target session
- Retrieve the primary access token
- Duplicate the primary token
- Copy environment variables
- Assign the working directory
- Create the process

Here's a library that I've pieced together, that abstracts the calls and ceremony needed to start a process in an arbitrary Window session, without handle leakage.

LeeWhite187/OGA.ProcessExtensions.Windows.Lib

It was adapted from ng-pe/cassia and murrayju/CreateProcessAsUser

See the repository page for example usage.

Revision #2
Created 8 January 2025 04:06:30 by glwhite
Updated 19 March 2025 15:39:38 by glwhite