

# Creating Transforms: Important *and* Easy

“Do not repackage MSI setups. Do not edit vendor-provided MSI setups.” You’ve probably heard these warnings a million times by now. One reason for the warnings is that there are possible hard-coded references in a MSI package that may be broken, but the biggest reason is that you will be prone to problems when the vendor releases an update or patch in the future. And forgetting the warnings altogether – there really is no need! You can make any change to an MSI setup quite easily (and not take on any of the risks involved with direct modification) using Windows Installer transform (MST) files.

There are two key reasons to employ transforms:

- 1) To automate the choices in the silent installation of an MSI setup
- 2) To customize a setup to change, remove or add to an existing MSI setup

## Response Transforms

With a transform, you may specify serial numbers, feature installation choices, target directories, or anything else you would normally enter during an interactive setup. If you have heard people talk about specifying property values using ORCA and said “forget it”, you are not alone. The good news is that there are tools out there that let you specify choices and values for an installation without the need to “edit” a solitary thing.

AdminStudio provides a “Response File Wizard” that lets you go through the steps of responding to a Windows Installer setup, recording your selections as you go. It takes you right up to the point where the files would begin installing onto your system and stops - that’s it, you have yourself an MST.

## Custom Transforms

If you want to delete a shortcut, add a file, modify a registry entry, or *anything* else at all, an MST can also do the trick. If you are repackaging a setup, it normally makes sense to just change these things in your setup directly. You could do that, right? Well doing so in an MST is nearly the exact same process. Simply say you want to create a new MST, tell it where your MSI is and the editor opens up just as if you were editing the MSI directly (see Figure 1 below). Make any and all changes you wish, hit save, and you have yourself a transform you can apply to make those changes “the right way”.

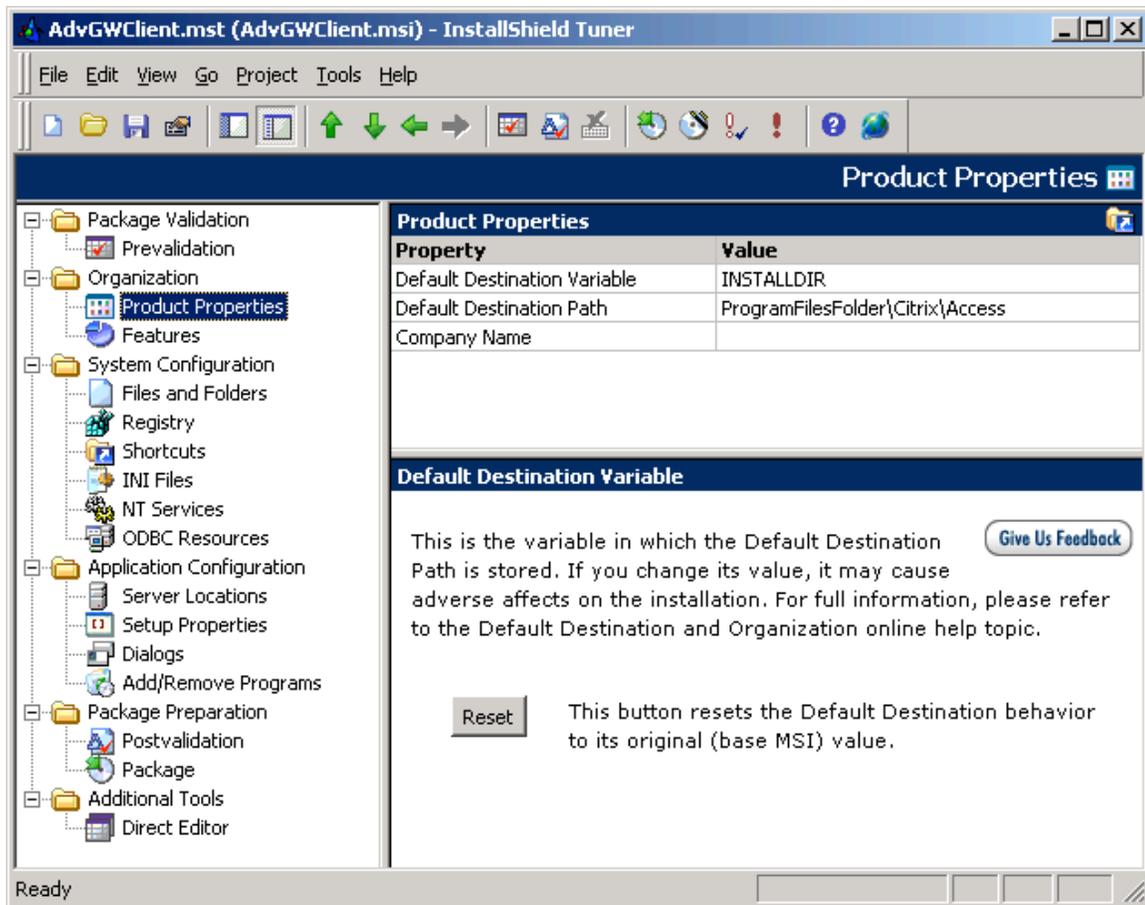


Figure 1: InstallShield Tuner (AdminStudio MST Creator/Editor)

And naturally, a combination of the two is as easy as creating your response MST, and then modifying it after the fact to make any further customizations you may require.

Group Policy introduces another key reason to use transforms, and that is the lack of an ability to specify the value of public properties (like you can on the command line). For example, if your target systems have multiple drive partitions software may not consistently be installed on the same drive. This is because Windows Installer will automatically install to the local drive with the largest amount of free space. Particularly if you have software that makes explicit calls to the "C" drive - or if the larger drive on a system is a removable one that is not always present - specifying a value for ROOTDRIVE is a good idea. Simply going into the property table and adding the ROOTDRIVE property with a value of C:\ will ensure the software is installed where you want.

Still others use transform files to institute minor changes to their own repackaged MSI setups between offices or business groups. The more you play with transform files and become comfortable with their use, the more you will see ways you might use them to your advantage within your own organization.