RELEASE NOTES

PACE Suite 4.5

Table of Contents

1 PACE Suite Overview

PACE Suite is an advanced and fast software packaging tool with an intuitive user interface and set of features providing (re)packaging in Windows Installer, APPX and various virtualized formats.

1.1 Key Advantages

- Out-of-the-box application packaging solution
- Intelligible wizards for running complex user scenarios
- High performance when generating large packages
- Per-user license valid for any number of physical and virtual machines

1.2 PACE Components

PACE Suite consists of the following components:

 MSI GENERATOR

- Repackage any existing installation into Windows Installer (MSI), UWP app package (APPX) or a virtualization format (Microsoft App-V 5.x, and VMware ThinApp).
- Auto-detection of embedded installers.
- Create a response transform
- Tune any existing Windows Installer package by applying your settings and saving them as a Windows Installer transform (MST).
- Include the excluded files/registry back to a package

 MSI EDITOR

- Manage the contents of an MSI in a tree view
- Import the necessary resources
- Edit and manage Custom Actions and Device Drivers
- Integrate scripts into your package
- Use a smart and advanced MSI database editor, with formatted string autocompleting, Excel-like formula bar, row reference tracking, and more.
- Create patches (MSP)
- Manage permission settings for file system and registry
- Handle upgrades easily – just let MSI Editor know which MSI you want to be upgraded at runtime.
- Publish your applications to Microsoft SCCM.
- See the estimated complexity of an app right away.
- Generate package documentation containing the details about your package and configuration.
- Undo-redo any manipulation and see the changes highlighted in the MSI tables

 DOCU GENERATOR

- Automatically record your on-screen actions into a nice-looking document with screenshots and annotations

1.3 System Requirements

Ensure that a target system contains necessary software prerequisites:

OS	Windows 10/8.1/8/7 SP1, Windows Server 2012/2008 R2 SP1 **NOTE** Building Universal Windows Platform app packages (APPX) is available under Windows 10 or Windows Server 2016 or newer.
Middleware	Microsoft .NET Framework 4.6.1 (https://www.microsoft.com/en-us/download/details.aspx?id=49981) or higher. **NOTE** For building ThinApp packages, ensure that your system contains VMware ThinApp (5.0.0/5.0.1/5.1.0/5.1.1/5.2.0) installed. **NOTE** For publishing MSI packages to Microsoft SCCM 2007 (SP1/SP2/R1/R2) or SCCM 2012 (SP1/SP2/R2/R2 SP1) server, ensure that your system contains Microsoft Management Framework 3.0 and Windows Remote Management (WinRM) service enabled.
Privileges	Administrative privileges on the system.

1.4 Hardware Requirements

Ensure that a target system fulfils the minimum hardware conditions:

Processor	32-bit (x86) or 64-bit (x64) processor at 2 GHz or greater **NOTE** To repackage 64-bit applications or create 64-bit App-V packages, use PACE Suite on a 64-bit Windows operating system.
RAM	2 GB
Hard drive free space	1 GB NOTE Additional storage is required for your projects and packages and depends on their size and quantity.
Display	1280 x 768 resolution or higher.

2 Release Notes

2.1 About This Release

In our new **PACE Suite 4.5**, we have focused on one of essential parts of software packaging - managing Custom Actions and Device Drivers. Our principal goal in this release was to simplify working with drivers and custom actions and provide enough guidance (especially through smart Wizards) so that even after complex customizations, our users will end up with packages that work.

First, we have created a separate Drivers tab, which lets you review, edit, and remove driver files, as well as apply various customizations like setting the installation options. The new "Add Driver" Wizard guides you through every step of including drivers into your packages and saves you time with useful tricks like automatically detecting and displaying all files that are related to your driver.

Next, we have completely redesigned the Custom Action tab, which now displays custom action details, allows moving custom actions to required sequence positions instantly, as well as highlights your changes and shows errors and warnings.

To simplify adding and editing custom actions, we have developed the "Add Custom Action" Wizard. The most exciting feature of our Wizard is that it takes into consideration your every input on every step and later disables options that are incompatible with your previous choices. Thus, "Add Custom Action" Wizard helps you ensure that your package will work properly in the end.
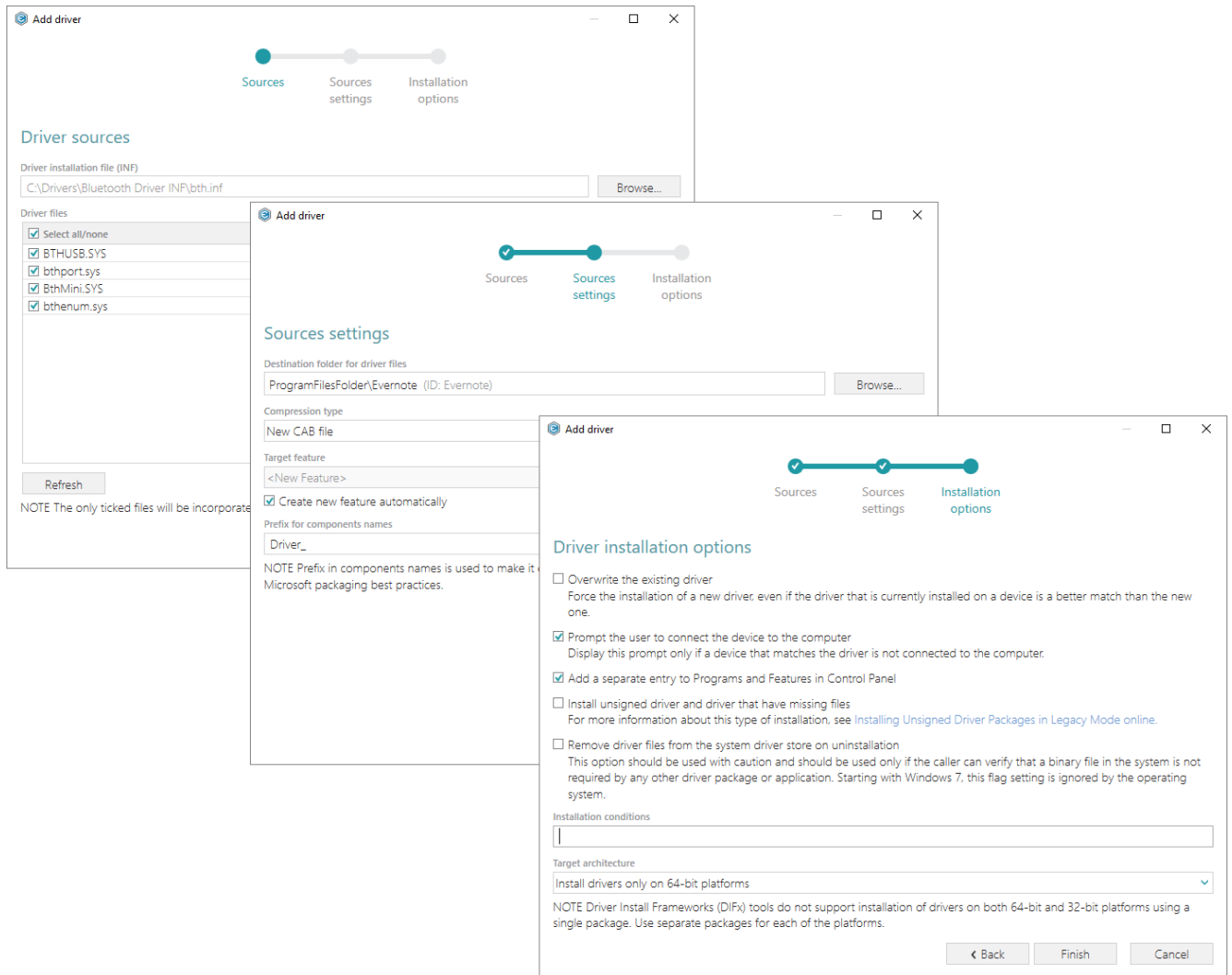
Finally, as usual, we have improved the stability of how several features work in especially tricky packaging scenarios.

2.2 Important Highlights

2.2.1 "Add Driver" Wizard

New "Add Driver" Wizard enables our users to add device drivers to their packages. You can choose driver installation files, and the Wizard automatically detects files that are related to your driver, and you can deselect ones that are unnecessary or missing from your file system.
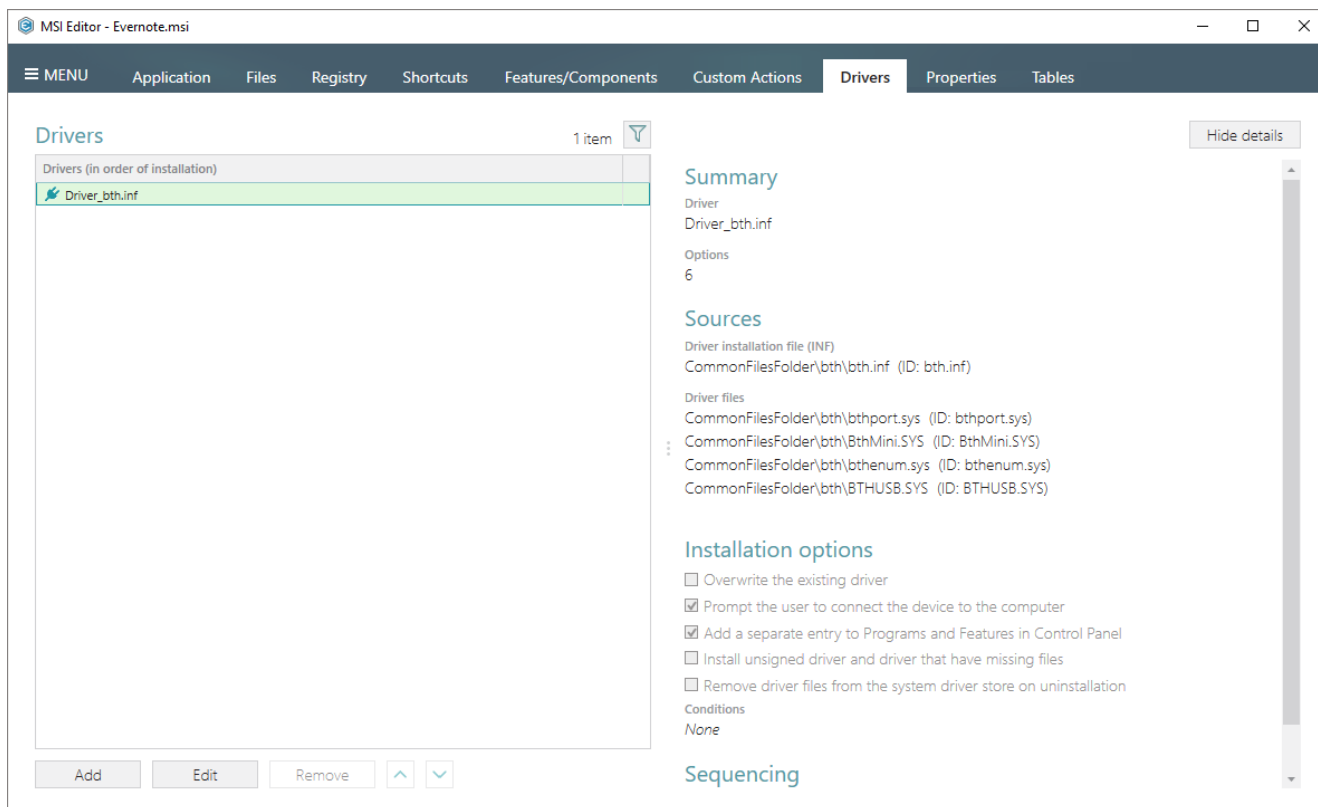

You can choose a folder in your MSI, to which driver files will be imported. For driver files, you can specify compression type and a feature, to which driver files will be assigned. Wizard also allows specifying prefixes in components' names so it will be easier to find driver components.

In the Wizard, you can set driver installation options, conditions, and choose target architecture (32/64bit). Drivers are installed on the system with the help of Driver Install Frameworks (DIFx) tools, a standard solution recommended by Microsoft.
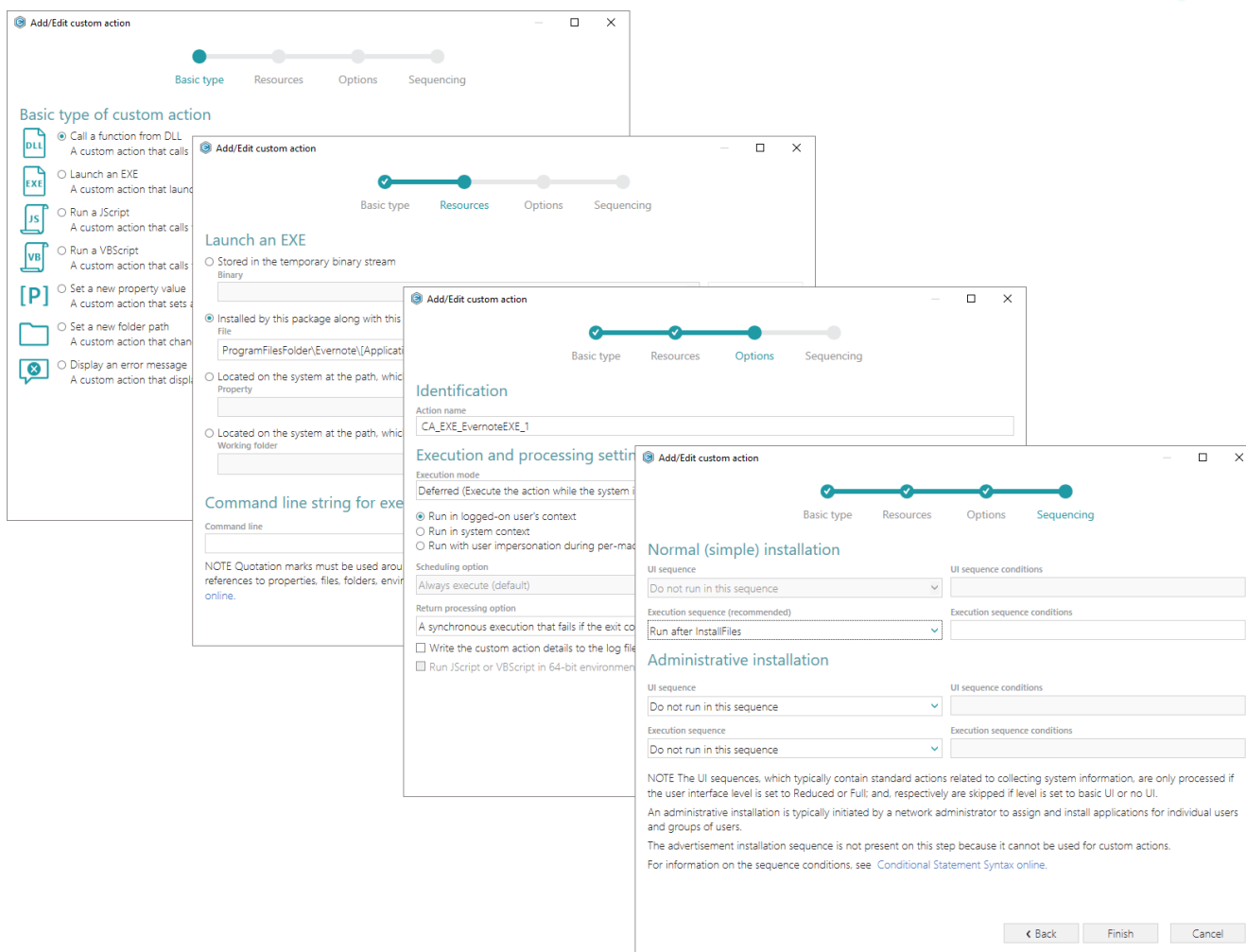
2.2.2 New Drivers Tab

We have added a new separate tab for Devices Drivers, where you can review, edit, and remove drivers. The tab shows details such as driver installation files (INF), files that are related to a particular driver, conditions, and installation options. You can set various installation options like overwriting an existing driver, adding a separate entry to Programs and Features in the Control Panel and others.

All the changes you make are highlighted, and validation errors (if any) are displayed. Note that the tab displays drivers that are installed using DIFx – a standard solution recommended by Microsoft.

2.2.3 "Add Custom Action" Wizard

Our new Wizard guides you through every step of creating a custom action. Depending on the basic type of custom action that you choose at the first step, the Wizard changes the contents of the next steps respectively.

When you choose resources for your custom actions, the Wizard displays only the resources that are related to the basic type you have previously picked. For instance, if the basic type is "Launch an EXE", the tree-like interface displays only EXE files and filters out everything unrelated.

Next, you set custom action options. All options, which are not compatible with your previous choices, are disabled.
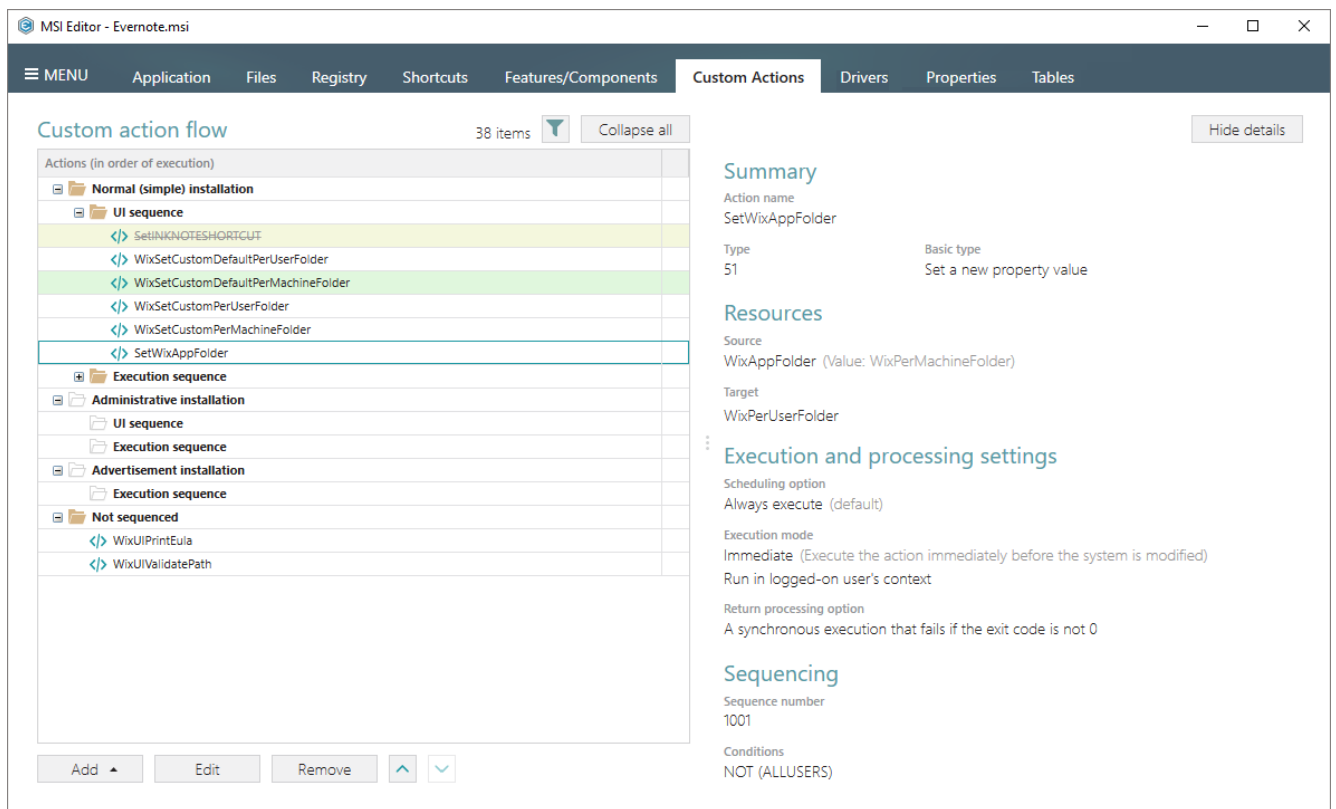
Finally, you can select a sequence position where your custom action will be run. Again, all positions, incompatible with your previous choices, are disabled.

In general, our Wizard takes into account all your inputs and helps you ensure that your custom actions will work.

2.2.4 Completely Redesigned Custom Actions Tab

We have completely redesigned the Custom Actions tab, which now displays custom and standard actions from all sequences. In this new tab, you can also find not-sequenced custom actions, which are present only in the CustomAction table and do not have references in other sequence tables.

You can review custom action details such as their basic type, scheduling options, execution mode, sequencing, and others, as well as find information about their resources.

In the tab, standard actions and custom ones have different icons so that you can tell them apart. Any changes you make are highlighted, and errors and warnings (if any) are displayed. If you need to edit your custom actions, the Wizard is again at your service with its guidance mechanism, described above.

You can add standard actions to each of the sequences. You can also move custom actions up or down in a sequence or use "Move to position" feature to put your custom action into the required sequence position (and this feature shows only positions that are available for your particular custom action).

2.3 Other Improvements

2.3.1 Reviewing Wise Package Studio project files

Some of our users still have their old Wise Package Studio project files with valuable information. Though you could have opened those files in PACE Suite previously, now you can open .WSI files with MSI Editor directly from the context menu and review their contents in case you want to use this information for creating new MSI packages in PACE Suite.

2.3.2 MSI Generator

We improved the approach MSI Generator uses to assign 32-bit and 64-bit registry to components in certain complex cases.

2.3.3 MSI Editor

We enhanced the overall stability of MSI Editor in the following cases:

- Using Active Scripting in particularly large packages.
- Rebuilding CAB files.
- Importing registry.

3 Release History