

RELEASE NOTES

PACE Suite 5.3







Table of Contents

| | | |
|-------|--|----|
| 1 | PACE Suite Overview | 3 |
| 1.1 | Software Prerequisites | 4 |
| 1.2 | Hardware Prerequisites | 4 |
| 2 | Release Notes | 5 |
| 2.1 | About This Release | 5 |
| 2.2 | Automated Testing | 6 |
| 2.3 | Automated Repackaging | 7 |
| 2.4 | Merge Modules Integration | 8 |
| 2.5 | Enhanced Exclusion Filters | 9 |
| 2.6 | Association Detection On Exclude | 10 |
| 2.7 | Improved Import Settings | 11 |
| 2.8 | Other Improvements | 12 |
| 2.8.1 | MSI Generator | 12 |
| 3 | Release History | 13 |

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, UWP (MSIX/APPX) and various virtualized formats.

Components:

| Tool | Major functions |
|---|--|
|  MSI GENERATOR | <ul style="list-style-type: none"> • Repackage any existing installation into Windows Installer (MSI), UWP app package (MSIX/APPX) or a virtualization format (Microsoft App-V 5.x, and VMware ThinApp). • Auto-detection of embedded installers. • Create a response transform • Include the excluded files/registry back to a package. |
|  MSI EDITOR | <ul style="list-style-type: none"> • Manage the contents of an MSI in a tree view • Import the desired resources • Manage Custom Actions • Integrate Device Drivers • Use a smart and advanced MSI database editor, with formatted string autocompleting, Excel-like formula bar, row reference tracking, and more. • Create patches (MSP) and Merge Modules (MSM) • 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 2007, 2012, 2016. • 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 |
|  QUALITY APPROVER | <ul style="list-style-type: none"> • Test MSI (+MST) package and its apps for compliance with the Desktop App Certification Program in an unattended mode. This will check the package safety, compatibility and readiness for deployment. For more information, see Windows Desktop app tests. • Test an MSIX/APPX package (bundle) and its apps against the Microsoft Store requirements in an unattended mode. For more information on the tests, see Windows Store App tests and Windows Desktop Bridge app tests. |
|  DOCU GENERATOR | <ul style="list-style-type: none"> • Automatically record your on-screen actions into a nice-looking document with screenshots and annotations |

1.1 Software Prerequisites

Ensure that a target system contains necessary software prerequisites:

| | |
|------------|---|
| OS | Windows or Windows Server OS that supports .NET 4.7.2 (https://docs.microsoft.com/en-us/dotnet/framework/get-started/system-requirements) or higher. |
| | NOTE Building Universal Windows Platform app packages (MSIX/APPX) is available under Windows 10 or Windows Server 2016 and newer. Automated package testing with Quality Approver requires a virtual machine with Windows 10 version 1803 or newer. |
| Middleware | Microsoft .NET Framework 4.7.2 (https://dotnet.microsoft.com/download/dotnet-framework-runtime) 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 and activated with the valid license. 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. |

1.2 Hardware Prerequisites

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

This new release is mainly dedicated to automation of package quality control. Now the user can easily test the package safety, compatibility and readiness for deployment. We have designed a new application - **Quality Approver** – for this purpose. Quality Approver runs locally and interacts with a local and remote pre-configured virtual machines (VM) for executing [Windows App Certification Kit](#). The Kit tests your MSI/MSIX/APPX package and its apps in an unattended mode.

We have introduced the support of command line options in **MSI Generator**. The use of command line options allows to automate the capturing (repackaging) process.

In order to eliminate the negative effect of system noise (log and cache files, registry entries, etc.) on the generated packages, we have reconsidered the pre-defined exclusion filters. The newly added filters allow detecting and filtering out a system garbage more efficiently. This saves the reviewing time and increases the package quality.

We have also implemented a number of feature requests from our customers and enhanced the performance of MSI Generator.

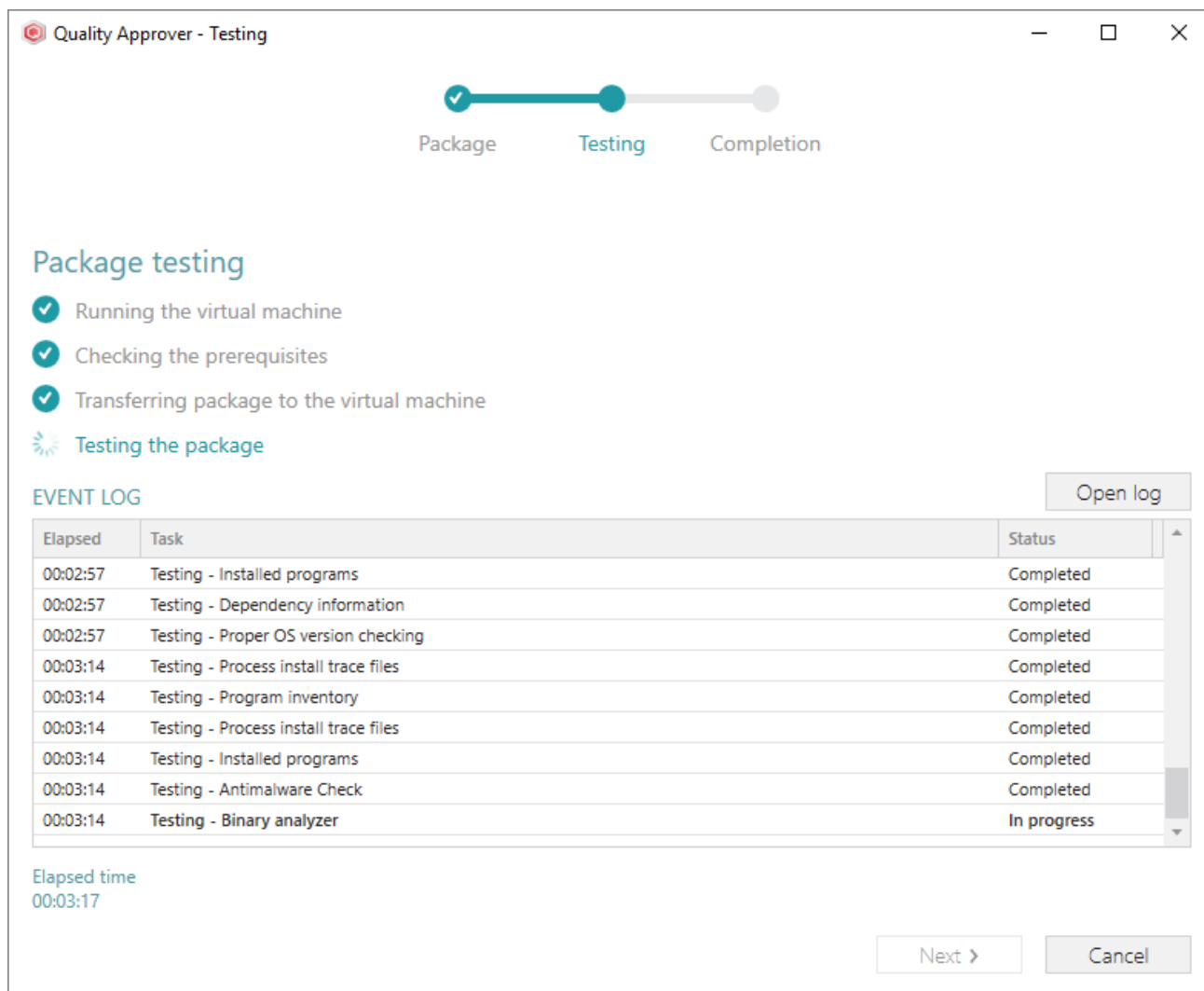
2.2 Automated Testing

Needless to say how important it is to check the package safety, compatibility and readiness for deployment. **PACE Suite - Quality Approver** is our answer to this challenge. **Quality Approver** is a tool that runs locally and interacts with either local or remote pre-configured virtual machines (VM) to execute the following tests.

- **Test MSI (+MST) package** and its apps for compliance with the [Desktop App Certification Program](#) in an unattended mode. For more information, see [Windows Desktop app tests](#).
- **Test an MSIX/APPX package (bundle)** and its apps against the Microsoft Store requirements in an unattended mode. For more information on the tests, see [Windows Store App tests](#) and [Windows Desktop Bridge app tests](#).

Quality Approver supports automated package testing on multiple virtual machines, hosted on the **VMware ESXi**, **Microsoft Hyper-V** and **Oracle VirtualBox** hypervisors. Quality Approver can run the tests locally as well, if the local host is configured as virtual machine which can be reverted to the previous state.

Quality Approver invokes the wizard which guides the user through the steps as shown below.



Release Notes

PACE Suite 5.3

Once the test completes, the user receives an HTML report with the results, the potential impact of issues and the suggested fixes. Here is the sample test report:

Microsoft Windows 10 Enterprise (10.0.17134.0)
OS Architecture:
x64
Report time:
26.07.2019 23:13:05

Overall result: PASSED WITH WARNINGS
It is highly recommended that you investigate and root cause warnings to ensure that users of your app aren't negatively impacted.

Clean reversible install test

PASSED
Remove all non-shared files and folders

PASSED
Do not force an immediate reboot during installation

PASSED
Do not force an immediate reboot during uninstallation

PASSED
Write appropriate Add/Remove Program values

PASSED
Single user registry check

Install to the correct folders test

WARNING
Install to Program Files

- Warnings: The install to Program Files test detected the following errors:

Program 7-Zip 19.00 fails due to install location C:\Temp\6daffe7b\12459029\.
- Impact if not fixed: Native apps e.g. 32 and 64 bit must be installed to the %ProgramFiles% folder by default, note that 32 bit apps installing on x64 versions of Windows must be stored under %ProgramFiles(x86)%
- How to fix: Write the files to the correct folders. See link below for more information:
[Working with Known Folders](#)

PASSED
Write to the %WINDIR% or %SystemDrive%

PASSED
Loading apps on Windows startup

Digitally signed file test

WARNING

2.3 Automated Repackaging

Automation saves time and efforts. PACE Suite introduces the command line support for **MSI Generator**. The use of command line options allows automation of the capturing (repackaging) process. The user can easily write a short script for running the pre-install and post-install scans with MSI Generator, apply the filters to the captured resources and generate the package as it is shown below.

```
:: Running pre-install scanning (taking first snapshot)
"C:\Program Files (x86)\PACE Suite\MSI Generator\MsiGenerator.exe" -snapshot "path\to\snapshot\first.msgn"

:: Running source installation
"C:\Source_installation\installer.exe" /silent

:: Running post-install scanning (taking second snapshot)
"C:\Program Files (x86)\PACE Suite\MSI Generator\MsiGenerator.exe" -snapshot "path\to\snapshot\second.msgn"
```

Release Notes

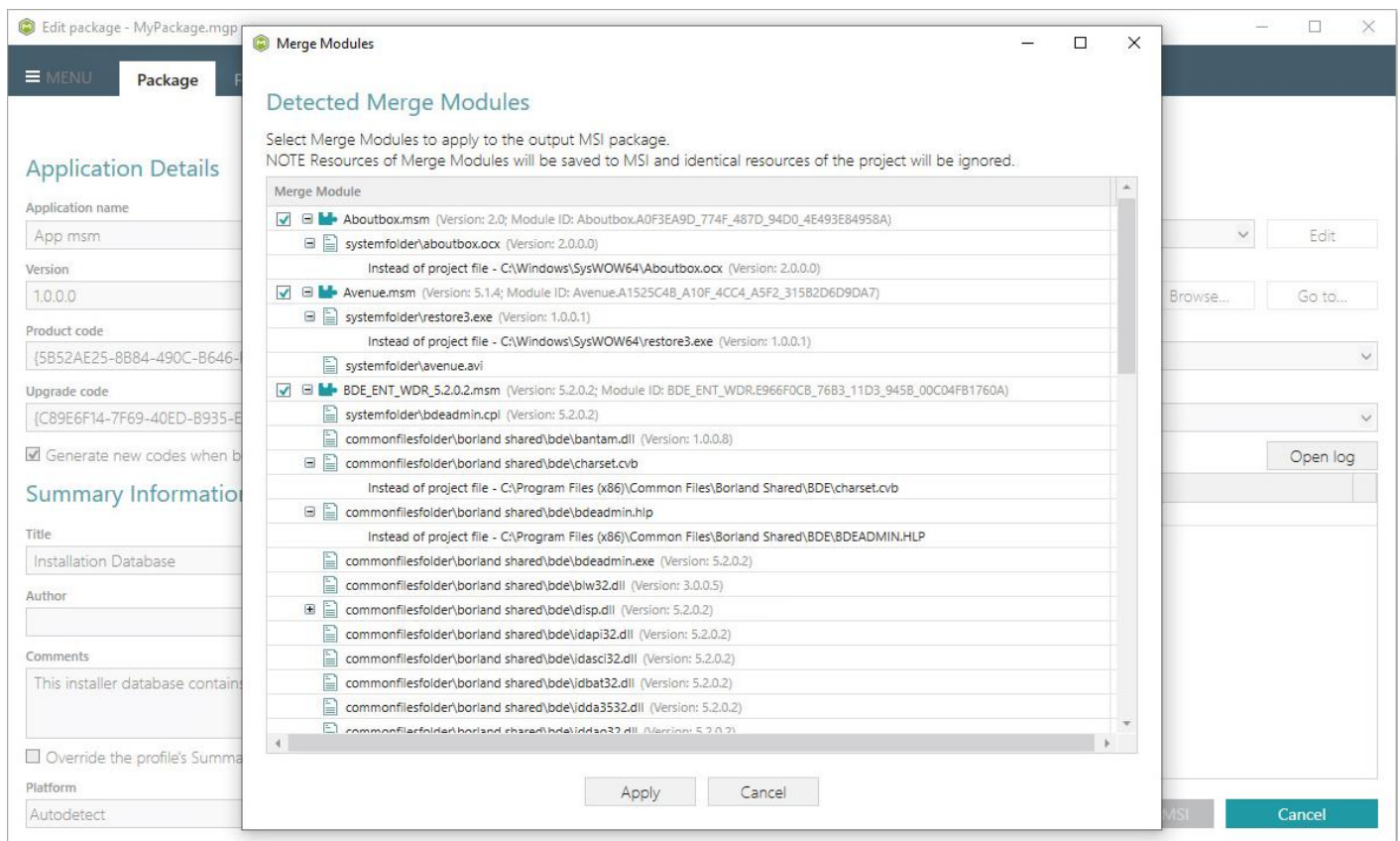
PACE Suite 5.3

```
:: Creating project (comparing first and second snapshots)
"C:\Program Files (x86)\PACE Suite\MSI Generator\MsiGenerator.exe" -compare "path\to\snapshot\
first.msgn" "path\to\snapshot\second.msgn" "path\to\output\project\folder" "projectName"

:: Building MSI package from project
"C:\Program Files (x86)\PACE Suite\MSI Generator\MsiGenerator.exe" -buildPackage "path\to\
output\project\folder\projectName\projectName.mgp" "path\to\package.msi" "msiSettingsProfile"
```

2.4 Merge Modules Integration

The use of [Merge Modules](#) (MSM) ensures a proper delivery of the shared code, files, resources, registry entries, and the setup logic to the installed applications. **MSI Generator** takes care of detection and integration of the appropriate Merge Modules into your repackaged installation.



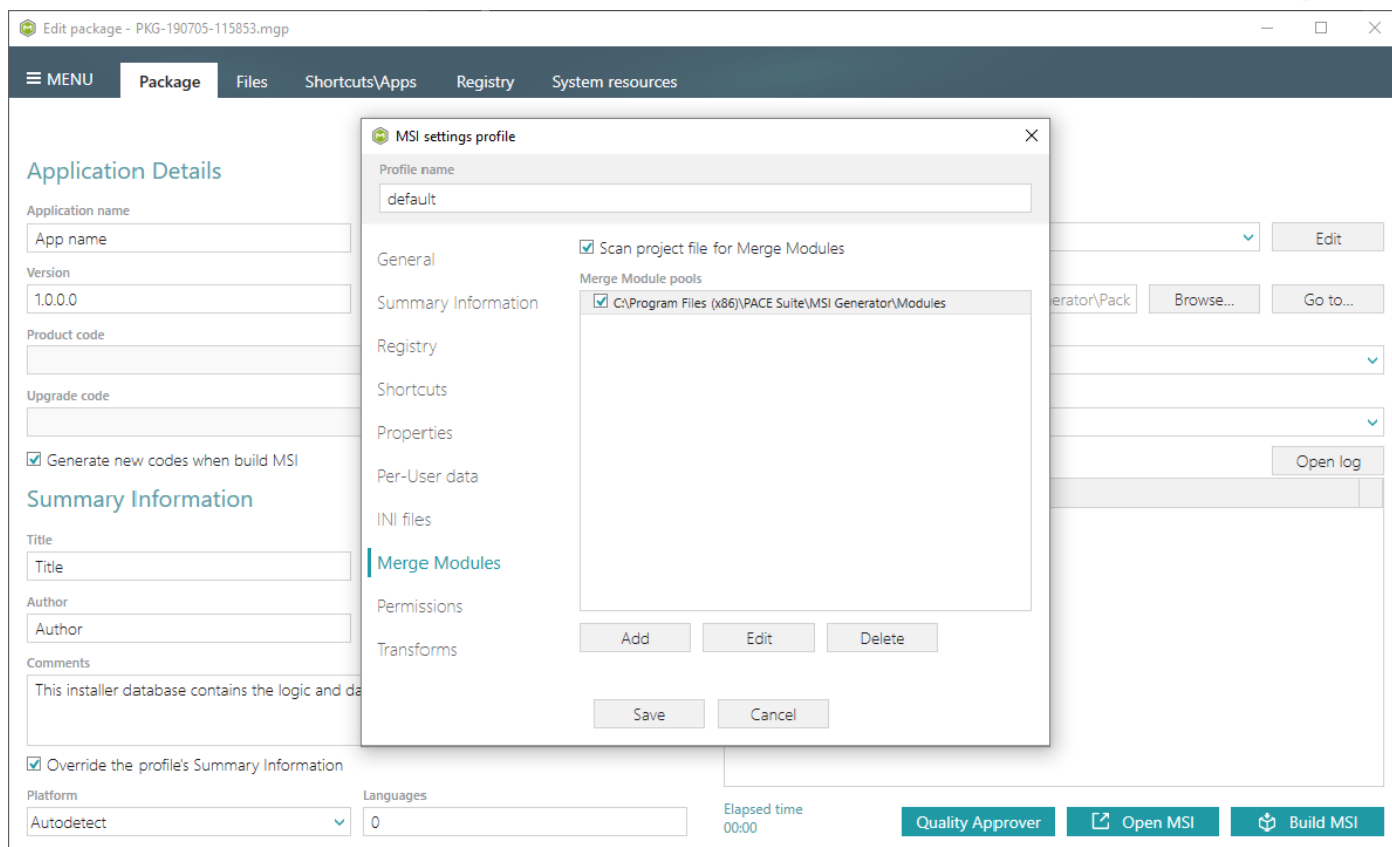
For the convenience, we have included the most common and recent Merge Modules into MSI Generator. You can find them at [C:\Program Files \(x86\)\PACE Suite\MSI Generator\Modules](#).

With the first installation of PACE Suite, this pool of Merge Modules will be automatically added to the default **MSI settings profile**. With update to PACE Suite 5.3, the user will have to edit the **MSI settings**

Release Notes

PACE Suite 5.3

profiles in order to add this pool manually.



2.5 Enhanced Exclusion Filters

Windows operating system and Windows services produce a lot of system noise (log and cache files, registry entries, etc.) that could have a negative impact on the quality of generated package. In many cases, an MSI package with a captured system noise cannot be installed as it tries to write to the system-protected locations.

We enhanced our pre-defined exclusion filters in **MSI Generator** aiming to filter out this system noise more efficiently. This saves the reviewing time and increases the package quality. The new filters will be applied to the captured resources by default.

Capture installation

Prepare computer Capturing method **Settings** Installation Completion

Settings

Package name
PKG-190731-152019

Exclusion filters

Apply the following filters to the captured resources:

- ☒ Extended
- ☒ Standard

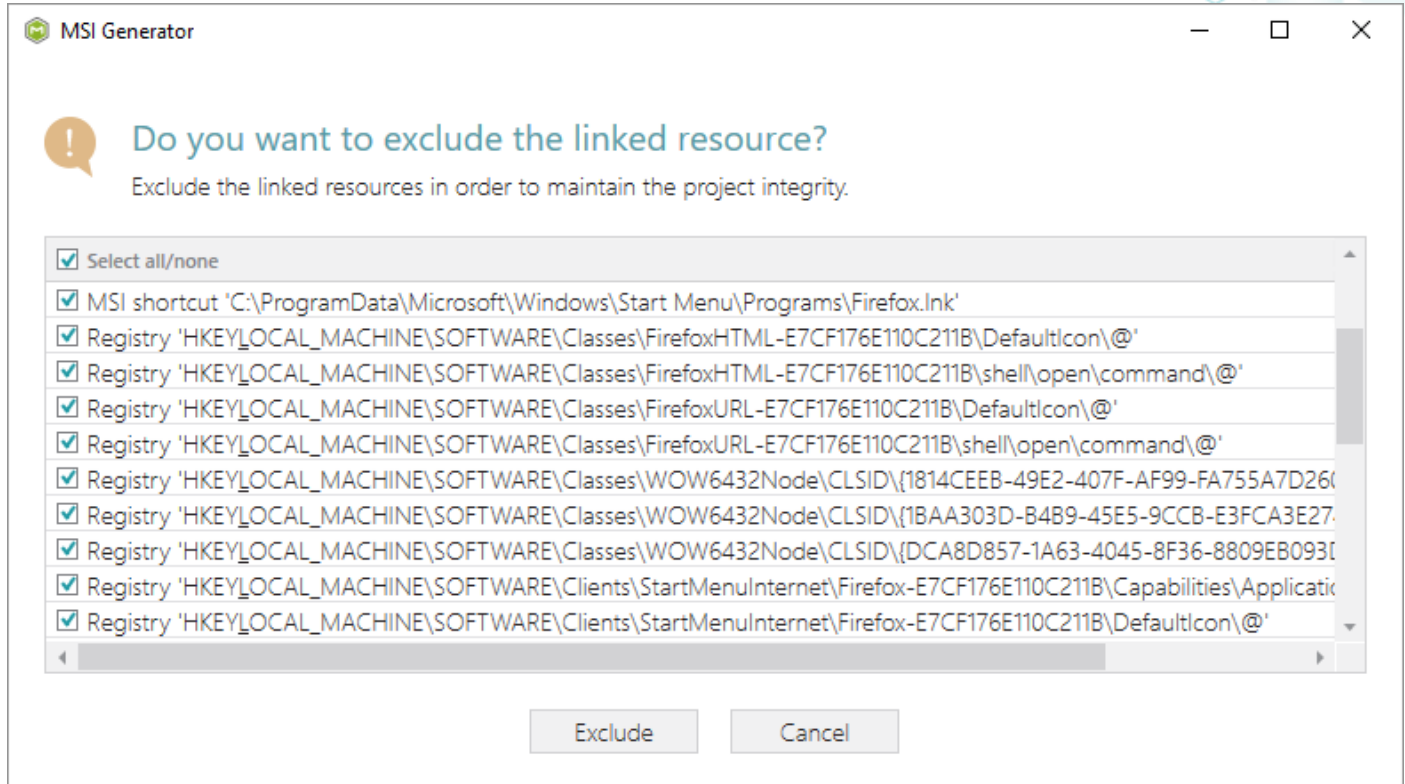
Scanning areas

| | |
|---|---|
| Registry: | Files: |
| <input checked="" type="checkbox"/> HKLM | <input checked="" type="checkbox"/> C: (Local Fixed Disk) |
| <input checked="" type="checkbox"/> HKCU | <input checked="" type="checkbox"/> D: (Local Fixed Disk) |
| <input checked="" type="checkbox"/> HKU\DEFAULT | <input checked="" type="checkbox"/> F: (Local Fixed Disk) |
| Application objects: | |
| <input checked="" type="checkbox"/> Permissions | |
| <input checked="" type="checkbox"/> Services | |
| <input checked="" type="checkbox"/> Printers | |

< Back Next > Cancel

2.6 Association Detection On Exclude

The user may like to exclude a captured library file (DLL, OCX, etc.) with all its registry entries. The newly introduced intelligent mechanism in **MSI Generator** detects and displays registration (COM) information, which is directly or indirectly associated with the excluded file. The user can either exclude the linked items or leave them included.



2.7 Improved Import Settings

PACE Suite 5.3 presents improvements to management of settings for importing files and registry into MSI packages. In the new version of **MSI Editor**, you can import resources to a package with the following feature-component assignments:

- **Create new features and components** according to the packaging best practices.
- **Create new components only** and link them with the selected feature.
- *(NEW)* **Use the existing component.**

Note that the use of the last option could imply certain restrictions on the imported resources. For instance, all imported files will be placed into a single folder defined by the selected component. The original folder structure will be lost in this case. Mixing per-user and per-machine registry entries in a single component may influence the Windows Installer self-healing mechanism.

The screenshot shows the 'Import files' dialog box with the 'Settings' tab selected. At the top, a progress bar indicates the current step is 'Settings', with 'Files' and 'Import COM' as previous and next steps. The 'Settings' section includes 'Feature-component assignment' with two radio buttons: 'Create new features and components' (selected) and 'Create new components only'. Below this is a 'Parent feature' text field containing 'Feature Title A (ID: FeatureA)' and a 'Browse...' button. Another radio button 'Use existing component' is unselected, followed by a 'Component' text field containing 'Component_A (Folder: ProgramFilesFolder\Folder A)' and another 'Browse...' button. A note states: 'NOTE All files will be imported into one single folder, defined by Component. Original folder structure will be lost.' The 'INI handling' section has three radio buttons: 'Save valid INI files to the 'IniFile' table; invalid ones to the 'File' table (recommended)' (selected), 'Save all INI files to the 'File' table', and 'Save only valid content of all INI files to the 'IniFile' table and then duplicate all INI files to the 'File' table'. A link 'See invalid INI files' is provided. At the bottom, there is a checkbox 'Complete now with default settings' and three buttons: '< Back', 'Next >', and 'Cancel'.

Import files

Files Settings Import COM

Settings

Feature-component assignment

☒ Create new features and components

☐ Create new components only

Parent feature

Feature Title A (ID: FeatureA) Browse...

☐ Use existing component

Component

Component_A (Folder: ProgramFilesFolder\Folder A) Browse...

NOTE All files will be imported into one single folder, defined by Component. Original folder structure will be lost.

INI handling

☒ Save valid INI files to the 'IniFile' table; invalid ones to the 'File' table (recommended)

☐ Save all INI files to the 'File' table

☐ Save only valid content of all INI files to the 'IniFile' table and then duplicate all INI files to the 'File' table

[See invalid INI files](#)

☐ Complete now with default settings

< Back Next > Cancel

2.8 Other Improvements

2.8.1 MSI Generator

- Added support of Japanese and other non-english typographic symbols.
- Decreased the UI response time for the Files, Shortcuts\Apps -> MSI, and Registry tabs.
- Added the calculated number of excluded resources on the Files, Shortcuts\Apps -> MSI, and Registry tabs.
- Added the filter for displaying the only excluded resources.

3 Release History

- [PACE Suite 5.2](#)
- [PACE Suite 5.1](#)
- [PACE Suite 5.0](#)
- [PACE Suite 4.7](#)
- [PACE Suite 4.6](#)
- [PACE Suite 4.5](#)
- [PACE Suite 4.4](#)
- [PACE Suite 4.3](#)
- [PACE Suite 4.2](#)
- [PACE Suite 4.1](#)
- [PACE Suite 4.0](#)
- [PACE Suite 3.5.5](#)
- [PACE Suite 3.5](#)
- [PACE Suite 3.4.2](#)
- [PACE Suite 3.4.1](#)
- [PACE Suite 3.4](#)
- [PACE Suite 3.3 – 3.3.1](#)