RELEASE NOTES

PACE Suite 6.1

Table of Contents

1 About This Release

This release of PACE Suite includes:

- Editor for PSAppDeployToolkit

  For details, see section 1.1

- Repackage in Windows Sandbox

  For details, see section 1.2

- Repackage in Docker

  For details, see section 1.3

- PowerShell in Custom Actions

  For details, see section 1.4

- Boost Performance of Remote Capturing

  For details, see section 1.5

- Improvements and Fixes

  For details, see section 1.6

1.1 Editor for PSAppDeployToolkit
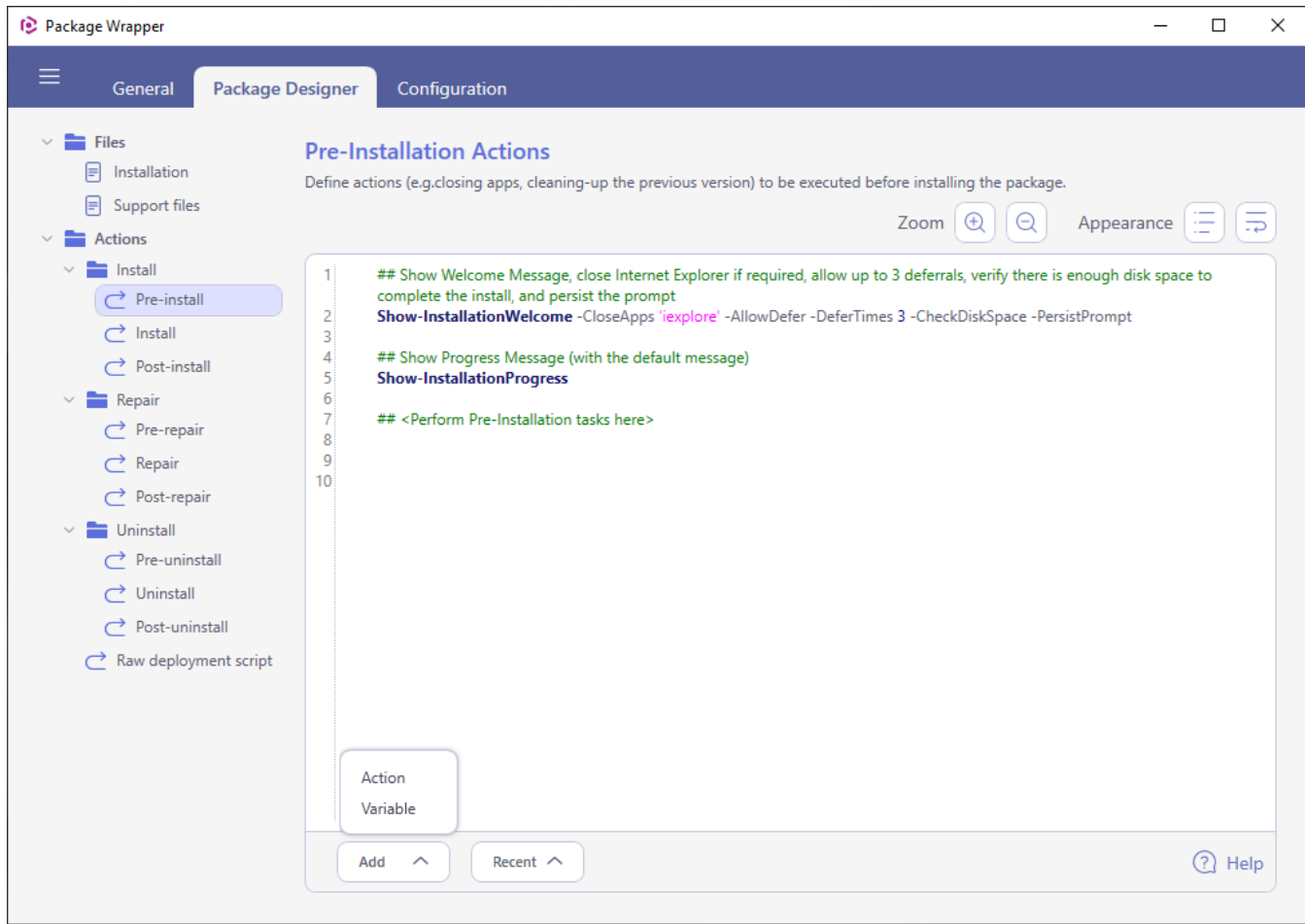
The **PowerShell App Deployment Toolkit** (PSAppDeployToolkit) is becoming increasingly popular as it allows you to deploy applications in a corporate environment without any additional effort.

To simplify the wrapping of your installations with the help of the PSAppDeployToolkit framework, we have developed **Package Wrapper** – a new tool in the PACE Suite family.

Highlights of Package Wrapper:

- Support of personal/team PSADT templates;
- Easy package creation of required PSADT version;
- Advanced script editor with syntax highlighting and row numbering;
- Validation of logical structure;
- Help with finding and adding actions and variables;
- UI for managing package appearance settings.
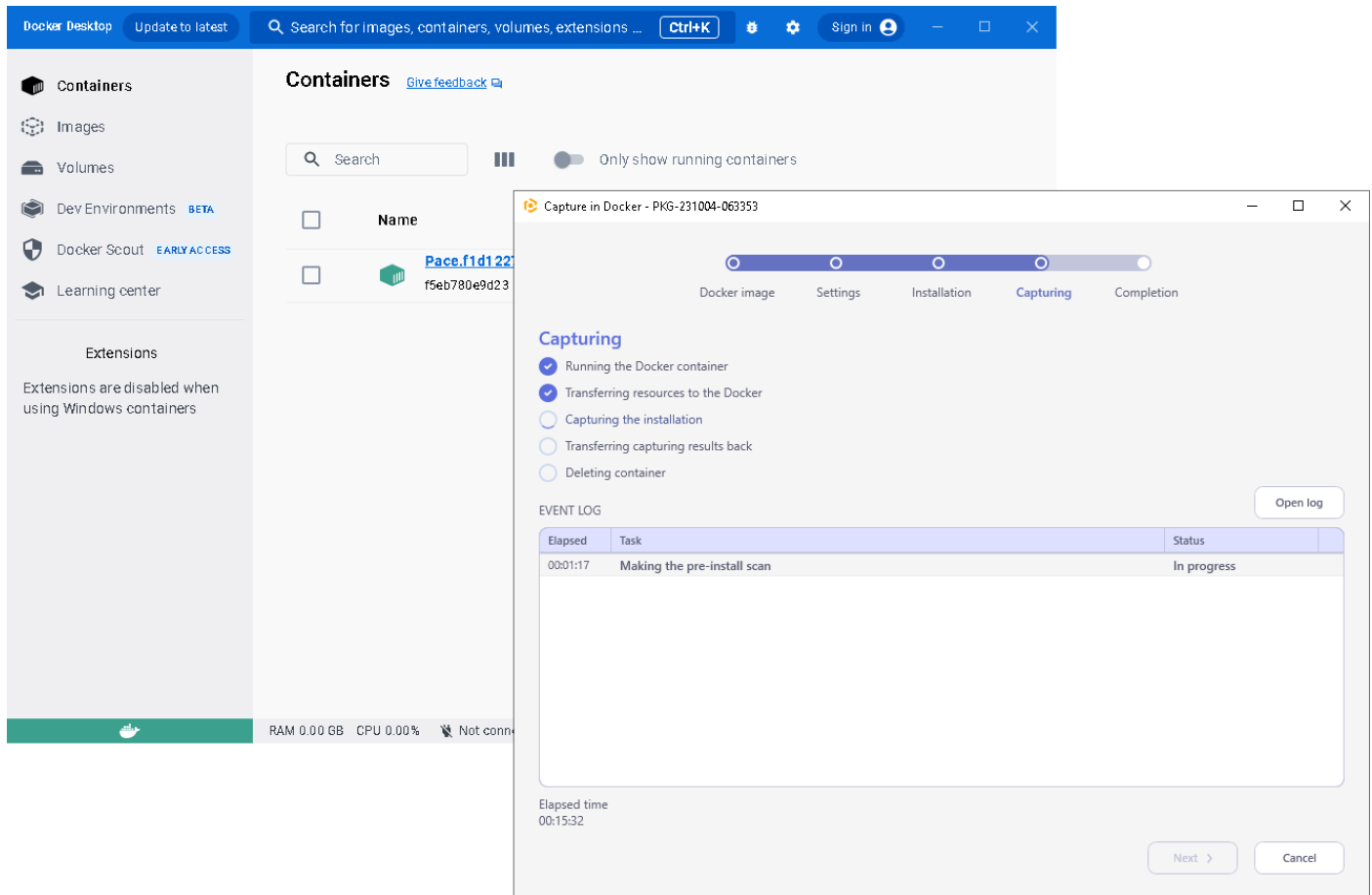
1.2 Repackage in Windows Sandbox
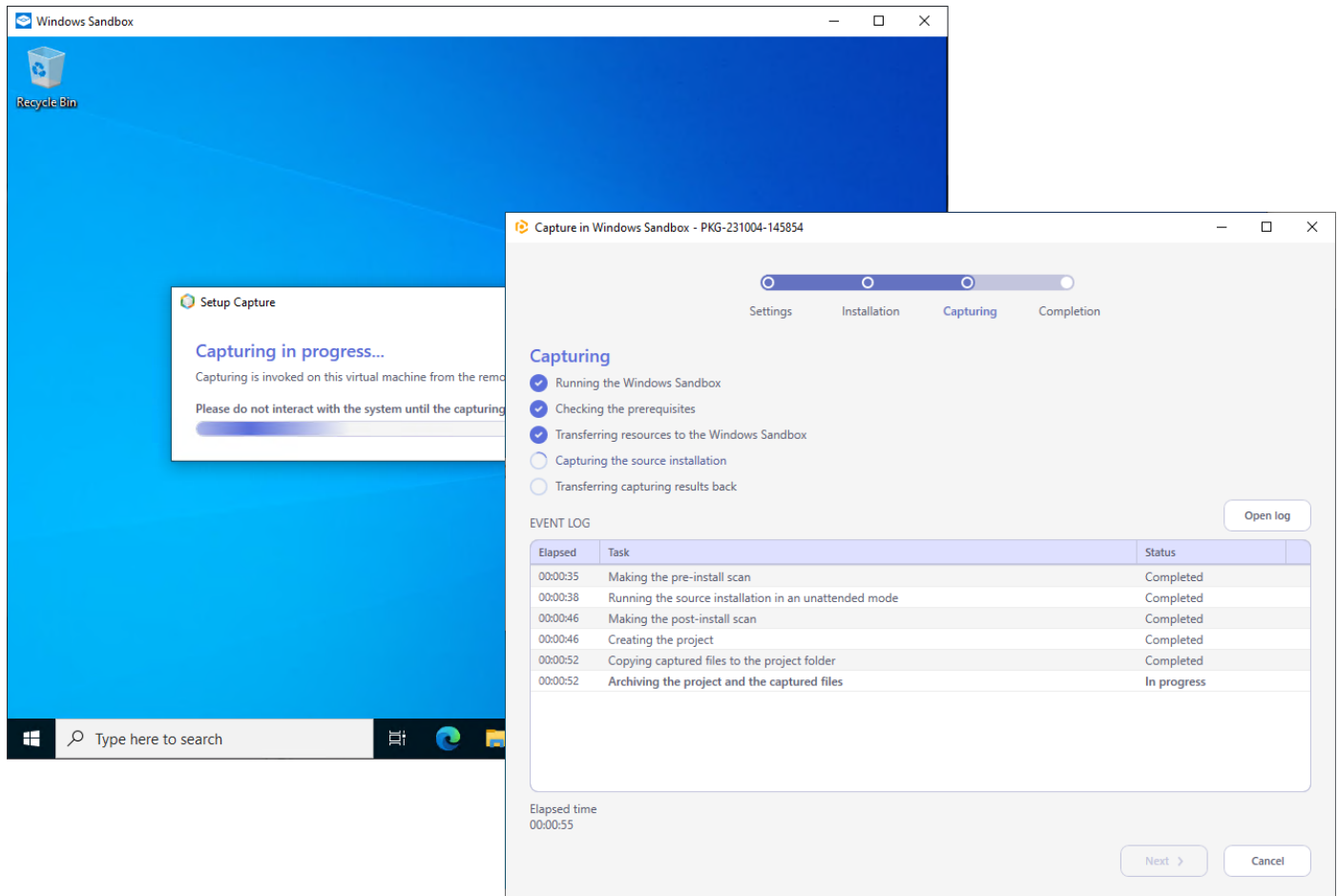
Windows Sandbox is a built-in Windows OS feature that provides a temporary desktop environment to safely run applications in isolation. Software installed inside the Windows Sandbox environment remains "sandboxed" and runs separately from the host machine.

We found this to be an ideal environment for capturing installations and therefore develop support for **Windows Sandbox** in our **Setup Capture**.

Advantages:

- Built-in Windows OS feature — no need to install and configure;
- Instant start and close of Windows Sandbox;
- No need to copy source and output files, as they are directly mapped;
- Support of manual and fully unattended capturing mode;
- In comparison to a regular VM, the sandbox is optimized for host resource consumption;
- The desktop environment in Sandbox does not require an additional Windows license;
- Provides a command-line interface (CLI) for automation.

1.3 Repackage in Docker

We have added support for capturing in **Docker** to our **Setup Capture**. The approach of **Repackaging in Docker** is identical to **Repackage in Windows Sandbox**, except that Docker Desktop must be installed and configured manually.

Docker provides an isolated Windows environment for repackaging that can be recreated for each capture from a single Windows OS image. Docker allows running multiple parallel instances of identical capturing containers.

In contrast to the benefits of repackaging in Docker, there are limitations, such as the lack of a UI session in Windows images. This means that you cannot interact with installer dialogs. Repackaging in Docker makes sense if you want to convert, for instance, MSI into MSIX or your installer supports silent mode.

With PACE Suite, you can repackage applications in Docker in a fully unattended mode and the command-line interface (CLI) provides automation abilities.

1.4 PowerShell in Custom Actions
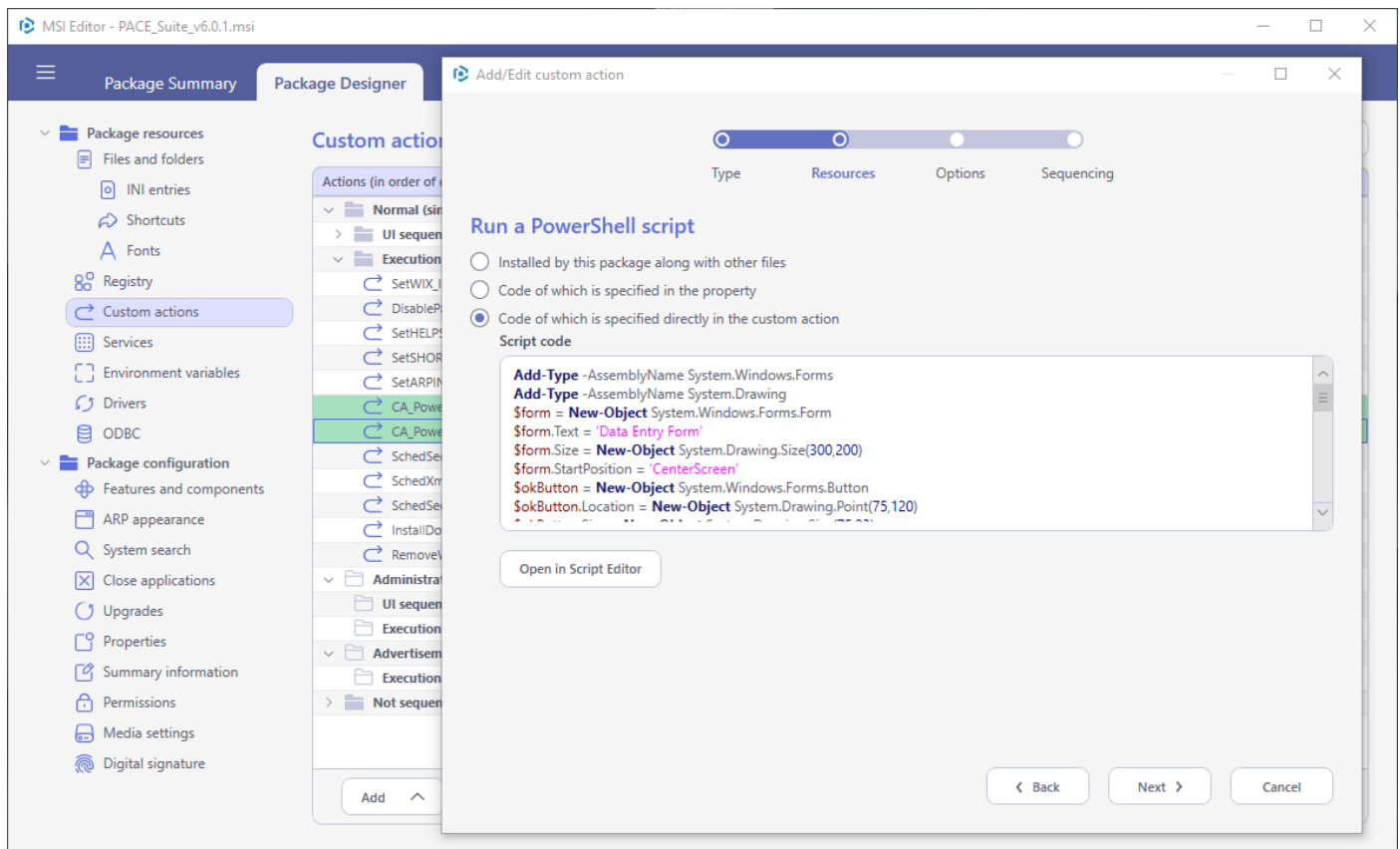
In this release, we have added support for PowerShell custom actions. Previously, such a task required several manual actions and had certain constraints.

Now, custom actions for running PowerShell scripts support three options for the location where the PS script will be saved. The storage location also affects the sequence position of the script run. We currently support running the script in the early stages of installer launch – right after the 'InstallInitialize' action.

```
Script editor                                              —  □  ✕

  PowerShell  ▾    History ⟳ ⟲    Zoom ⊕ ⊖    Appearance ☰

  1   Add-Type -AssemblyName System.Windows.Forms
  2   Add-Type -AssemblyName System.Drawing
  3   $form = New-Object System.Windows.Forms.Form
  4   $form.Text = 'Data Entry Form'
  5   $form.Size = New-Object System.Drawing.Size(300,200)
  6   $form.StartPosition = 'CenterScreen'
  7   $okButton = New-Object System.Windows.Forms.Button
  8   $okButton.Location = New-Object System.Drawing.Point(75,120)
  9   $okButton.Size = New-Object System.Drawing.Size(75,23)
 10   $okButton.Text = 'OK'
 11   $okButton.DialogResult = [System.Windows.Forms.DialogResult]::OK
 12   $form.AcceptButton = $okButton
 13   $form.Controls.Add($okButton)
 14   $cancelButton = New-Object System.Windows.Forms.Button
 15   $cancelButton.Location = New-Object System.Drawing.Point(150,120)
 16   $cancelButton.Size = New-Object System.Drawing.Size(75,23)
 17   $cancelButton.Text = 'Cancel'
 18   $cancelButton.DialogResult = [System.Windows.Forms.DialogResult]::Cancel
 19   $form.CancelButton = $cancelButton
 20   $form.Controls.Add($cancelButton)
 21   $label = New-Object System.Windows.Forms.Label
 22   $label.Location = New-Object System.Drawing.Point(10,20)
 23   $label.Size = New-Object System.Drawing.Size(280,20)
 24   $label.Text = 'Please enter the information in the space below:'
 25   $form.Controls.Add($label)

                     Apply        Cancel
```

If you need to view/modify the script, you can do that right in the Custom Actions Wizard with the help of the Script editor. This editor provides syntax highlighting, row numbing, and zoom options.

1.5   Boost Performance of Remote Capturing
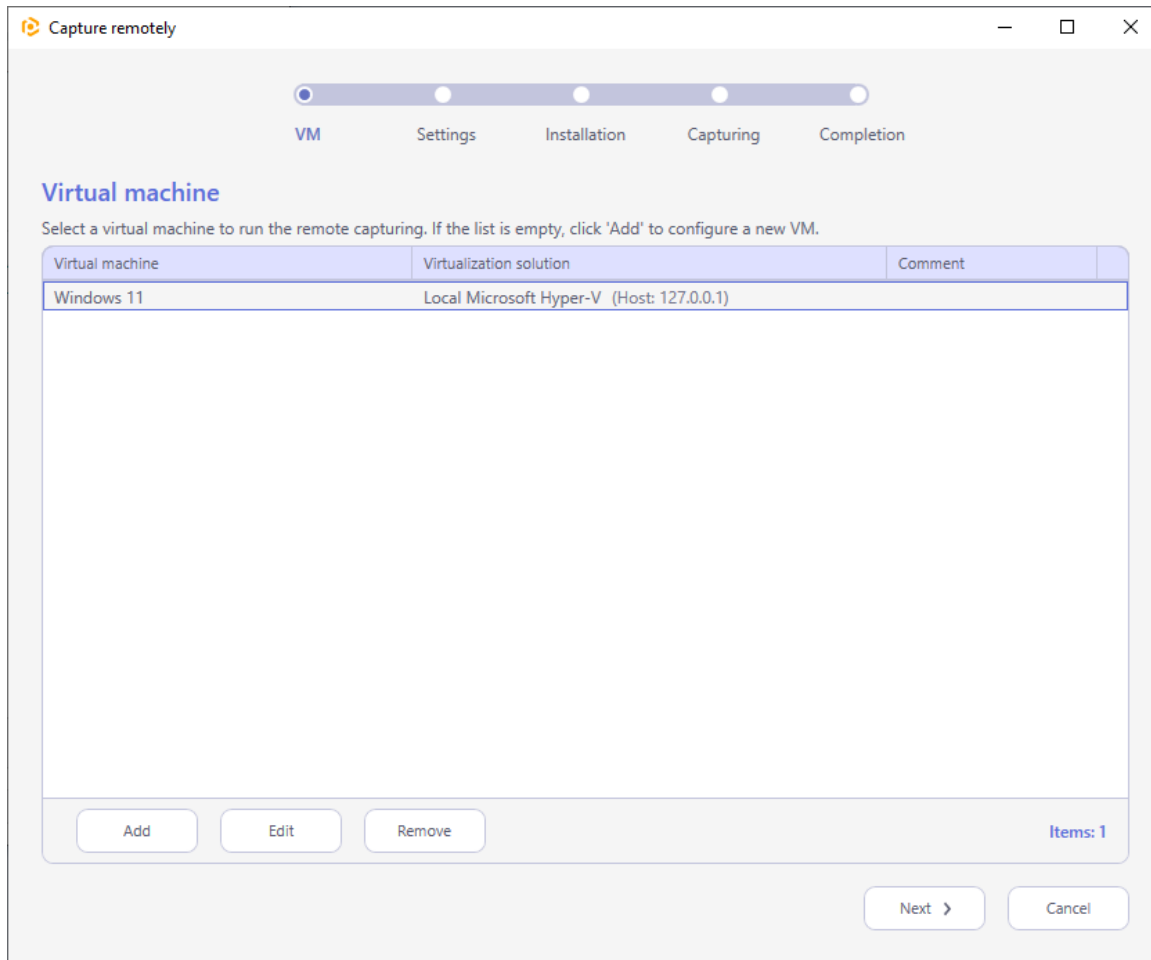
We've redesigned the approach of archiving resources that are transferred to and from a remote virtual machine during the repackaging process. According to our tests, the new algorithm and multi-threading increased performance by 4 times.

We hope this improvement will allow you to do more in less time.

1.6   Improvements and Fixes

This release includes numerous fixes to the quality of created packages, data handling, data display, highlighting of changes, and more. Most of the reported bugs and stability issues were fixed.

2   Release History

- [PACE Suite 6.0](#)
- [PACE Suite 5.8](#)
- [PACE Suite 5.7](#)
- [PACE Suite 5.6](#)