# Custom Package Creator for Parallels Client for MacOS

## Summary

This tool allows administrators to easily create a custom Parallels Client for macOS install package for deployment at scale. It uses an exported Parallels Client configuration file (.2xc) from an already configured Parallels Client for macOS used as a blueprint. Parallels Client for macOS is publicly available at https://www.parallels.com/products/ras/download/links/

It allows administrators to preconfigure common client settings during the deployment which is useful when deploying clients at scale.

The custom package can then be delivered at enterprise scale with tools such as Jamf or other Mobile Device Management (MDM) or Enterprise Mobile Management (EMM) solutions with the required changes in place ahead of users opening the Parallels Client for the first time.

## Download the tool

To obtain the latest copy of the tool, follow this link to our public GitHub repository <https://github.com/Parallels/RAS-PowerShell/tree/master/Tools> and download the contents of the folder titled “RAS Custom MacOS”

The core instructions for use are shown below and are also included in the file README.md which is in this folder

Within this download is a folder called RAS with several sub-folders and two bash scripts within. One called 'createpackage' and the other called 'postinstall'

A screenshot of a computer

AI-generated content may be incorrect.

## Tool contents

The 'createpackage' script takes a standard Parallels Client for MacOS install and the custom .2xc file you export (see below) and creates a custom .pkg file to be used in mass deployment tools such as Jamf

The 'postinstall' script is incorporated into the final custom package and runs after the Parallels Client for MacOS has been deployed and is what delivers the customisations the administrator has saved in the .2xc file

## Preparation

### Creating a .2xc file

* Install the Parallels Client for MacOS (the current tool has been tested up to client version 20.2.1 but should also work with subsequent versions, though full testing is advised prior to mass deployment)
* Configure the client settings and connections that are needed using this client install as a template configuration.
* In the Parallels Client for MacOS, Navigate to File - Export Settings to export a .2xc file with those settings configured.  
  A screenshot of a computer

  AI-generated content may be incorrect.
* If editing is needed (for example to remove the username used in testing) you can open this .2xc file with a text editor and customize
* To be used by this tool the 2xc file you save needs to be in the following folder: RasClientDeploy > Scripts > Contents. The tool looks in this folder for a file ending .2xc.

IMPORTANT make sure there is only one 2xc file in this folder as the script will use the first .2xc file it finds

### Download the notarized Parallels client for MacOS

* Download the latest Parallels Client for MacOS here: (https://www.parallels.com/products/ras/download/links/) - or otherwise use the version you wish to deploy
* To be used by this tool the .pkg file you download above needs to be saved in the following folder: RasClientDeploy > Scripts > Contents. The tool looks in this folder for a file ending .pkg.

IMPORTANT make sure there is only one pkg file in this folder as the script will use the first .pkg file it finds

## Creating the Custom Install

* Once both the .2xc file and the .pkg install are located within the folder: RasClientDeploy > Scripts > Contents - you are ready to create the custom package  
  A screenshot of a computer

  AI-generated content may be incorrect.
* Open a Terminal window at the root RasClientDeploy folder.
* Run the 'createpackage' script by entering the following command ./createpackage
* A file called ParallelsClient.pkg is created within the root folder of the tool  
  A screenshot of a computer

  AI-generated content may be incorrect.
* You are now able to mass deploy your custom Parallels MacOS Client package using tools such as Jamf by adding this file ParallelsClient.pkg as a package in such tools

## Example Deployment (Jamf Pro)

To assist in the deployment of the custom package, the steps to deploy in Jamf Pro are shown below. Other deployment approaches using alternative services are broadly similar.

## Note, in the process below is an example only, the steps were performed using Jamf Pro version 11.18.1. The steps needed in future versions may differ.

IMPORTANT: because this package imports custom initial configuration, it should only be used for initial setup. The native client process or another package should be used for ongoing update to avoid overwriting the configuration for a second time

### Add the Package to Jamf Pro

In the Jamf Pro console, navigate to Settings > Computer Management > Packages

A screenshot of a computer

AI-generated content may be incorrect.

Once in Packages – select New and give the package an appropriate name. Then either drag or browse to the custom .pkg file you created earlier. This saves the file to the default distribution point.

A screenshot of a computer

AI-generated content may be incorrect.

Once your package is saved, navigate to Policies > New Policy provide an appropriate name for this policy and a trigger for when you’d like it to apply and whether you want it to be available in self-service for the user.

You can also select the scope here to determine which computers or users to make this policy available to.

Then Navigate to Packages and select the Configure button and choose the package you uploaded above but clicking on the Add button on the right.

A screenshot of a computer

AI-generated content may be incorrect.

When you have finished editing the Policy, choose Save.

Once the Policy is saved, it is ready to go and will deploy according to the settings you have specified.