Mac Packaging Standards

Each package provided by TEIS will endeavor to meet the following packaging standards. These standards aim to surpass the expectation of users, providing them with the most positive experience possible when utilizing software installers created by this team.

# Naming Conventions

Packages provided by TEIS should conform to the following naming conventions.

## File Name

Package files should be named with their vendor and title with capitalization as specified by the vendor, with a default of the leading character of each capitalized. The vendor and title should not be separated by any space of punctuation in the file name.

* **<Vendor><Title>**\_<Version>.pkg

## Version

The version number should be separated from the vendor and title with an underscore ( “\_” ) character, and should contain the full version number (with all punctuation).

* <Vendor><Title>\_**<Version>**.pkg

## Revision Number

If multiple packages are released for the same version of a specific software title, a revision number shall be added as a suffix to the file name as to indicate which release the file is associated with. This can either be an incremental number (so that the second release would be numbered “r2”), or a secondary version number of a supplemental component that is being installed as a prerequisite.

* <Vendor><Title>\_<Version>**r<Revision>**.pkg

## File Extension

Files should be presented as OS X Installer Packages with the ‘.pkg’ file extension.  
Under **no** circumstances should the meta package (.mpkg) format be used.

* <Vendor><Title>\_<Version>**.pkg**

# File Format

Packages provided by TEIS should meet the following formatting requirements.

## Flat Package

All packages shall be presented as single flat files. This ensures compatibility with more file systems and deployment methods.  
“Bundle” or “Meta” packages are NOT to be used under any circumstances. Any packages provided by external vendors that are not flat files will require repackaging.

## Distribution Package

Each package intended to be executed by a user shall be packaged as a distribution package, as to provide a custom user interface that is consistent with the spirit of this standards document.

## Disk Image

Software should be deployed as a disk image (‘.iso’, ’.img’, or’.dmg’ file) only when specific requirement necessitate it. This format leaves an additional file to be removed by the user, and provides a less natural installation flow.

# User Interface

As distribution packages, all software installations developed by TEIS should contain the following user interface elements.

## Package Title

Each package should have a title that contains the vendor’s name and the title of the software, using the spacing and capitalization as provided by the vendor. This title should not include a specific version number, so as to allow package projects to be leveraged for future versions.

## Background Image

All packages should have a custom background, as approved by TE MarComm, to easily differentiate internally-created packages from vendor-supplied versions.

## Welcome Screen

The Welcome screen of each package should contain a clear list of minimum requirements, both as provided by the vendor as well as internal requirements based on policy.

## Template Text

This package will install <App Name> as provided for Mac users at TE Connectivity.

You will need to be sure you meet the following minimum requirements to install and use this software:

* Any Mac with a 64-bit Intel Processor
  + Intel Core 2 Duo or newer recommended
* <RAM Requirement>
* <HD Space Requirement>
* OS X 10.9.5 or later, as installed by TEIS
* TE Connectivity complaint local password
  + or AD-based local account
* McAfee Endpoint Protection

Click the 'Continue' button to proceed with this installation.

## ReadMe Screen

The ReadMe (Important Information) screen of each package should contain any special alerts the user needs to be aware of, as well as any important information provided by the vendor.

## License Screen

The Licensing screen of each package should contain the software license agreement for the software to be installed, as provided by the vendor.

## Success Screen

The Installation Summary (Success) screen of each package should contain an image (.png) representation of the software being installed’s icon, so as to provide easy identification by the user post-installation.

# Requirements

All packages provided by TEIS should check for systems to meet the following minimum requirements.

## OS Version

All packages provided by TEIS will check for the current, or next most current (e.g. “N & N-1”) version of the OS.

Dialog Text

* Before installation of this package can continue, you will need to update the version of OS X.
* Please refer to [the Apple Rolodex](http://www.apple.com) for more information.

## Platform

All packaged provided by TEIS will check that they are being installed on the currently supported Mac platform.

Dialog Text

* This installation can only be completed on 64-bit Intel-based Mac computers.
* Please refer to [the Apple Rolodex](http://www.apple.com) for more information.

# Package Identifiers (receipts)

Each component package must have a unique package identifier. These identifiers are used both as receipts (bill of materials listing all files installed by each package, and their permissions), as well as in displaying custom install options during the installation process.

## Vendor-supplied packages

Component packages provided by the vendor, even when modified by TEIS, should endeavor to retain the package identifier of the vendor’s package.

## TEIS-created packages

When appropriate, supplemental packages created by TEIS should mimic the formatting of vendor packages they are associated with.  
Otherwise, TEIS packages should use the com.te.connect.<title> nomenclature

# Component Packages

The following component packages are commonly found within TEIS distribution packages.

## Payload-less Packages

Some distributions may require additional scripting through the use of packages that execute scripts without deploying files (one example of this might be writing a TE license into an application’s preference file).

* The ‘—nopayload’ flag may be used on these packages to avoid creation of a package receipt on the local system.

## Application Installation

These are packages, either provided by the developer, or created/customized by TEIS, that deploy the primary application payload to the end user’s machine. The version number of this package often sets the version number for the distribution package.

This component’s inclusion in a custom installation action commonly triggers other packages as being required during that custom installation.

## Supplemental Installation

These are packages, either provided by the developer, or created/customized by TEIS, that deploy additional resources or associated libraries & applications to the end user’s machine.  
While the version number of these packages are sometimes referenced in the revision number of the distribution package, their version number often varies (and is not directly related to) the version number of the primary component. These packages can often be added or removed at the user’s discretion during the custom install action.

## Uninstall & Removal

This is commonly a payload-less package that executes a removal script to revert the user’s system to a pre-installation state.  
The specifics of this script vary greatly from distribution package to distribution package, typically requiring custom scripting by the packager.

This package is commonly not included in a standard installation, but provided as an optional component for use during the custom install action.

## Internal Customizations

These are supplemental installation packages, typically developed by the packager, that customize the installed application for compatibility with the TE production environment (the most common example of this is a TE-specific license).  
This can take the form of additional packages added to the installation, or customizations to the vendor-provided packages (such as additions to the post-install script).

# Logging

Every package shall log all actions taken for later review in both the QAT process, as well as any troubleshooting efforts (by the user, the Mac community, or a service professional).

## Location

All packages should use the same location for ease of locating log files.

/Library/Logs/TEIS/

## File Name

A shortened, single word (no spaces), name or keyword, unique to each software title.  
To be universal to all versions of the package, and all subsequent versions of the software title, so should not contain any version information.

* /Library/Logs/TEIS/<SoftwareTitle>.log

## Scope

Each separate action being taken by the installer, other than placing files, needs to be explicitly logged, with individual entries per action.  
No log activity should occur when no action should be taken. This allows for scripts to be executed as needed, without excessive log chatter. Generally speaking, log failures, not successes.  
Where natural, STDOUT or STDERR output of individual commands can be redirected to the log file in lieu of discreet logging actions.

## Formatting

Messages should be logged in the following format:

* [Date] [User] [Package] [Section] Text
* Date
  + The date and time that the action was taken.
  + This should be recalculated on each action, and should NOT be a global value for the time the script was executed.
  + All logging should utilize the following date format:
    - YYYY Mon DD hh:mm:ss
* User
  + The user who initiated the installation that executed the script performing the action being logged.
  + If installation is initiated by an automated process, any active console user at the time of process initiation.
* Package
  + The name of the package that executed the script performing the action being logged.
* Section
  + The name of the component, script, or sub-routine that is performing the action being logged.
* Text
  + Custom text that describes the action being taken.
  + This text should be consistent with other log messages, thus written in the third person, as phrases, with as little punctuation as needed to convey the action taken.

# Scripting

Scripts included in packages provided by TEIS should conform to the following guidelines.

## Language

Whenever possible, all package scripts should only utilize scripting languages native to the OS (such as BASH or AppleScript), and included in limited OS environments (such as the netboot environment, or the environment in recovery mode). Languages that are not native to these environments (such as Ruby, Python, & PERL) should be avoided unless specifically necessitated by individual packaging requirements.

## Common Variables

The following are variables commonly used by scripts bundled with installation packages created by TEIS.

SoftwareTitle

* One word (no spaces) title to be used a the name of the log file associated with this package.
* This should be consistent across all scripts used in all component packages for the distribution package.

SectionTitle

* Often set throughout the script to notate individual sub-routines, mainly for logging purposes.

LogFile

* The file where actions taken by the installation software should be logged. Used as part of the writeLog function below, as well as a redirect location for various commands throughout the script.
* /Library/Logs/TEIS/$SoftwareTitle.log

ConsoleUser

* The active user at the time of the package being executed.
* Used in the creation of user-specific file paths (such as “/Users/$ConsoleUser/ ...”), as well as to be sure that there is an active console user (not root) when performing specific actions (such as unloading launch agents or other user- specific libraries)
* $(stat -f %Su '/dev/console')

writeLog

* Common function to customize log writing, containing custom date formatting, as well as the user initiating the action, and the script (and sub- routine) that originate the log message (so that individual log messages can be forwarded without needing to reference the file name that contained them).
* { echo "[$(date "+%Y %b %d %T")] [$ConsoleUser] [$ScriptTitle] [$SectionTitle] $1" >> "$LogFile"; }

ResourceLocation

* Used in previous scripts to determine the file location where the script was being executed.
* For packages, this should resolve to the “Resources” folder within the package hierarchy, used to execute other files added during package creation (such as nested vendor packages).
* $(dirname "$0")

InstallationVolume

* Variable that resolves, in scripts for installation packages, to the target installation volume.
* Used in path creation to allow for installation onto a target volume other then the currently booted volume, such as in a netbook environment.
* $3

ReceiptList

* Command to list all package identifiers considered to be receipts of currently install packages associated with the current installation (change "$AppName" to a keyword of the application being installed). Mainly used during removal scripting to remove the package receipts forecasting installations.
* $(pkgutil --packages | grep “$AppName")

## Template

#!/usr/bin/env bash

### HEADER

# Package Title:

# Author:

### DEFINITIONS

## VARIABLES

SoftwareTitle=

SectionTitle=Postflight

LogFile="/Library/Logs/TEIS/$SoftwareTitle.log"

TimeStamp="$(date "+%Y %b %d %T")"

ConsoleUser="$(stat -f %Su '/dev/console')"

ResourceLocation=$(dirname "$0")

PackageName='.pkg'

## FUNCTIONS

writeLog(){ echo "[$(date "+%Y %b %d %T")] [$ConsoleUser] [$SoftwareTitle] [$SectionTitle] $1" >> "$LogFile"; }

### BODY

## LOGGING

[[ -d "$(dirname $LogFile)" ]] || mkdir -p -m 755 "$(dirname $LogFile)"

[[ -e "$LogFile" ]] || touch "$LogFile"

[[ "$(stat -f%z "$LogFile")" -ge 1000000 ]] && rm -rf "$LogFile"

## EXTRACTION

SectionTitle=Extraction

if [[ -e "$ResourceLocation/$PackageName" ]]; then

writeLog "Executing $PackageName"

installer -dumplog -verbose -pkg "$ResourceLocation/$PackageName" -target "$3" -allowUntrusted >> "$LogFile"

else

writeLog "File Not Found: $PackageName"

ls $ResourceLocation >> "$LogFile"

fi

## FOO

SectionTitle=

echo foo

### FOOTER

exit 0

# Codesigning

Packages provided by TEIS should conform to the following code signing requirements.

## Packages

All packages developed or edited by TEIS should be signed using TE Connectivity’s developer certificate. This is not currently possible given status of TE Apple Developer account.

## Applications

All applications deployed through packages provided by TEIS should be signed, either by TE Connectivity or by the vendor.

# Misc

Some additional guidelines for the behavior of packages provided by TEIS are:

## Dock

No package provided by TEIS should forcibly modify the user’s dock by default (though options that modify the dock may be provided through the custom install action).

## Pre-Installation Actions

Unless expressly required (such as commonly needed for application removal during uninstall actions), the user’s environment should not be modified (such as by quitting applications in use) or actions be taken that can adversely affect the user’s overall experience.

## Post-Installation Actions

Unless expressly required, the user’s environment should not be modified (such as windows opened), user’s preferences be modified, or actions taken that can adversely affect the user’s overall experience.

## User-specific Preferences

If a preference is not currently set at time of installation, the installation may modify the default setting of that preference, but preferences already set by the user should not be modified unless specifically as part of a security or policy requirement.