2.2.1.2 Installability

Return to Portability

- ☒ **[nick][✓ char, 2020-10-22]~~Char review done~~**
- ☐ **[DDSFmember]Please Review**

About

Return to the Top

installability and its mirror requirement of un-installability allows for an individual product or an entire system to be installed and uninstalled on a system easily, efficiently and with a minimum number of side effects. A side effect occurs when removing a product or system has adverse effects on other products or systems running on the same computer or network. For example, when a product was installed, it installed a library to access a particular dbms but, in so doing, removed a library required to access a different DBMS. These side effects can also occur when a product or system is un-installed. For example, when a product is un-installed, it removes a library used to access a particular DBMS; however, this library is required by other products or systems on the computer or network.

In addition, the complexity of installing or un-installing a product or system may be extremely tedious, detailed, and laborious with many steps and decision points. Each step in this process introduces risk. It is important to remember that risk is multiplicative, thus as the number of steps involved increases, even if the individual risk of each step is low, the overall risk to the success of the installation or un-installation increases.

Many of these issues were the bane of new systems and products in the past. Fortunately many of these issues have been addressed through the use of Wizards, packagemanager tools, containerization processes and orchestration tools. Installation usually entails the following checks and functions:

- The target system has the correct system resources available
  - Hardware architecture (32-bit or 64-bit cpu etc)
  - os such as Android, IOS, Linux, MacOS, Windows, Unix, etc.)
  - Network connectivity
  - Hardware resources such as memory, disk space, etc.
2. The target system does not already have the software installed (i.e., previous or current version of the software). If so, perform an upgrade.
3. The target system has the proper directories, files and operating system privileges
4. Add configuration data (i.e., configuration files, envvar, or windowsregistry entries) to the target system
5. Make software accessible on the target system (i.e., setting privileges, creating links, and shortcuts)
6. Configure components required to run the target system (i.e., daemon, services, etc)
7. Activate software on the target system (i.e., license agreements, license activation, system registration, etc.)

8. Update other third party components to acceptable levels on the target system

Installers can be classified according to the amount of interaction with the user:

Table 1: Installer Classifications

Kind of Installer	Description
Attended installation	This generally requires a user to attend the installation process to answer questions about where to do the install on the target system, provide information about the user, accept any terms and conditions, etc. [1)]
Silent installation	This allows installation of a system or program on a target system without any notification to the user. This is often the backdoor used by malware. It differs from **Attended** and **Unattended** installations only in that the user may be completely unaware of the installation.
Unattended installation	This installation is similar to the **Attended Installation** but does not require any human interaction, thus allowing for systems or programs to be installed with the user just monitoring the installation rather than interacting with it.
Headless installation	This refers to the graphics head usually used to drive a monitor. If there is no graphics head, there are only command line interfaces and logs for detailing the installation process. Often these installations use Telnet where the installation on one computer (or machine) is being done from another computer (or machine). **Note:** could be a virtual machine.
Scheduled or automated installation	This term is usually used with **Attended** or **Unattended** installation. The installation process is scheduled for a later time, or on a schedule (i.e., every first Tuesday of the month).
Clean installation	This installation can be considered “hostile” in that no regard is made for previous installations. It can also mean that it does not care about other systems or programs that are installed. It `makes clean` and then starts the installation process anew.
Network installation	This kind of installation is made using a shared network resource. It often requires the installation of a minimal or skeleton operating system before the rest of the installation can occur. This kind of installation is used often for large corporate, government or other institutional organization.

[1)] **Note:** It is possible to have hybrids of these kinds of installers. For example, an installer is **Attended** for the first part of an installation and then unattended for the remainder of the installation

In addition to the kinds of installations. Installers programs can either be **self contained**[1)] and specific[2)] in what they can install (i.e., **wizard**) or they can be generalized to handle any kind of installation (i.e., **packagemanager**). Sometimes, installer programs themselves need to be installed or updated, a pattern referred to as "bootstrapping", which entails the use of a lightweight, simple and small executable file that is initially downloaded and started on the target system. This executable updates the installer software and then launches into the installation of the desired software. Often in complex systems or programs, the initial bootstrapping process updates other components that the desired software depends upon. For example, an operating system or DBMS update.

[1] **Note:** Contain all the files they need to perform the installation.
[2] **Note:** product based such as Microsoft Word, Parallels

DDS Specifics

Return to the Top

data_distribution_service_dds is a midware product, and as such, is not generally installed by a user directly, but is integrated into other systems or products. The installer for the desired systems or products (i.e., Wizard or Package) are responsible for installing the correct DDS shared library. In Unix or Linux, a shared object library file is given the extension of **.so**. In Windows, the shared library is referred to as a Dynamic Link Library with the file extension of **".dll"**. There very few installation or de-installation issues when using DDS.