

osFree distribution (draft I)

osFree distribution must use:

- [WarpIN](#) as a package management tool, with enhancements. WarpIN's purpose is maintaining the installed packages database, understanding the .WPI format, installing/deinstalling each package
- Also, a tool like eCoShop or Linux-like repository access client is needed. It can be like apt or yum in Linux, but with OS/2 specifics. This tool must be separate from WarpIN, i.e., it is specific to repositories/package sources, not packages format. So:
 - it must handle different package sources, like
 - files in the current directory
 - special directory layouts (on the current machine, or on remote http or ftp server), like diskette images with bundles (as used by IBM OS/2 distributions so, the installer will be capable to install old IBM OS/2 distributions as well)
 - other directory layouts, like APT or RPM repositories
 - different file transport protocols support, like ftp, http, etc, via plugins
 - maybe, support for handling the different packages formats, other than WPI, i.e.:
 - plain ZIP's with metainfo and installation scripts included (like it was in UnixOS/2)
 - RPM - for software ported from UNIX.
 - pack/unpack-created bundles, as usual with IBM's OS/2 distributions
 - the package source frontend will handle different package types via special plugins (or backends), and invoke different tools when installing packages (invoke WIC, UNZIP, UNPACK or RPM)
 - maybe, even, the possibility to create the decentralised network of package repositories is needed – this will allow to use any nearest mirror, or use another mirror if the main mirror is down. Even we could try to create the repositories network on Peer-to-Peer basis ;)
- The enhancements to WarpIN should include:
 - “Sticky” flag for packages which should not be upgraded
 - Support for simultaneously existing versions of different packages with libraries, which are needed for different applications
 - Maybe, “Unstable” flag for apps and libraries, so, unstable package don't replace stable ones, they only create a separate branch in dependencies tree.
 - Coexistence of several package trees, which are updated separately, and do not influence other trees – some analogies with source code repositories with branches
 - Support for separating and merging the branches, rollback of packages to an older (but stable) version
- response-file-driven installer:
 1. response file can be created manually
 2. response file can be generated by UI (VIO or PM-based)
- so, installer must be divided into four separate parts:
 1. UI for choosing user options and generating the response file interactively, it can be:
 - textmode UI with pseudographics
 - graphical, PM-based
 2. installation engine, based on invoking the package access frontend to retrieve the packages. It acts according the response file
 - it can be started by an experienced user, system administrator, or by the installation UI.
 - installation engine is independent from UI and can be started separately; and more:

- the user can start the UI on one machine, save the resulting RSP (response) file and apply it on another machine, by pointing the installation engine to the needed RSP
- the single installer for initial machine setup, and subsequent option changes/additional software installation – something like FreeBSD sysinstall
 - also, DOS (DPMI)-based version of the installation engine and UI would be desired
 - a Linux- or Win32-based version (?)
3. package access frontend, handling different package sources, via backends:
 - different package storage/repository types
 - different access protocols are implemented as special plugins
 4. enhanced WarpIN as the main package handling tool.

From:
<https://cocorico.osfree.org/doku/> - **osFree wiki**

Permanent link:
<https://cocorico.osfree.org/doku/doku.php?id=en:docs:distribution&rev=1366484874>

Last update: **2013/04/20 19:07**

