



Safeguarding the
Future of Linux
Through Standards

An Overview of the Linux Standard Base (LSB)

Ian Murdock
Chief Technology Officer
Linux Foundation



Safeguarding the
Future of Linux
Through Standards

The LSB is...

- The Linux Standard Base (LSB) is a set of programming interfaces guaranteed to be present across compliant Linux platforms
- More simply: The LSB is the Linux “meta-platform” standard
- Basic idea: applications written to the LSB can run on any compliant distribution (i.e., no need to have different “ports” for each distribution)
- The Linux Foundation offers certification program around the LSB interfaces
 - Distributions
 - Applications
 - More to come



Safeguarding the
Future of Linux
Through Standards

The LSB is *not*...

- The LSB is not a distribution
 - We specify *interfaces*, not implementation; however, we go beyond previous standards (e.g. POSIX) by specifying binary behavior
- The LSB does not define new standards
 - We track best practices as they emerge in the open source community and document them
 - We work with key Linux platform stakeholders (distro vendors, ISVs, upstream developers, OEMs) to help drive consensus on key issues



Safeguarding the
Future of Linux
Through Standards

Value proposition

- The LSB is not a “write once, run anywhere” panacea!
 - We seek to augment, not replace, existing platforms, certification programs
- Big guys: Risk mitigation, greater portability, access to the “long tail” of applications and distributions with minimal additional investment
- Little guys: Access to applications/distributions that would be cost-prohibitive to support otherwise



Safeguarding the
Future of Linux
Through Standards

What makes a good platform?

- Uniform, stable, consistent, backwards-compatible API/ABI
 - Are my platform investments protected?
- Predictability
 - When will the next version of the platform be released and what features will it have?
- Broad enough scope
 - Does the platform provide the functionality my application needs?



Safeguarding the
Future of Linux
Through Standards

What makes a good platform (cont.)?

- Developer friendly
 - Tools
 - Documentation, central source of information for developers (“Developer Network”)
 - Learning curve (ISV: “Linux is terrifying to new developers, particularly those coming from the Windows or Macintosh platforms”)



Safeguarding the
Future of Linux
Through Standards

Platform thinking

- Alignment with roadmaps of major distros
 - Major version of LSB for every major version of enterprise distros
 - LSB 3.x is current “generation” of enterprise distros (SLE 10, RHEL 5, etc.); LSB 4.x is next generation (SLE 11, RHEL 6, etc.); etc.
 - Must be possible for end users and ISVs to translate between LSB and distro releases
- Backward compatibility beginning with LSB 3
 - ISVs can target a version of the LSB with assurance that their applications will not break in future LSB versions



Safeguarding the
Future of Linux
Through Standards

LSB roadmap

LSB 3.x	LSB 4.x
Asianux 2.0 Debian 4.0 (“etch”) Mandriva Corporate 4.0 Red Hat Enterprise Linux 4 and 5 SUSE Linux Enterprise 9 and 10 Ubuntu 6.06 LTS (“dapper”) [...]	Asianux 3.0 Debian “etch” + 1 Mandriva Corporate 5.0 Red Hat Enterprise Linux 6 SUSE Linux Enterprise 11 Ubuntu LTS “dapper” + 1 [...]



Safeguarding the
Future of Linux
Through Standards

Platform thinking (cont.)

- Direct involvement of upstream developers, ISVs and OSVs in LSB development process
 - LSB as “anticipating standard”
 - LSB as umbrella and integration point for other standards
 - LSB as collaborative forum for key stakeholders in the Linux platform
- LSB SDK, application testkit, LSB Developer Network to make it “dead simple” to build portable Linux apps using the LSB standard



Safeguarding the
Future of Linux
Through Standards

Project organization

- LSB is primary standardization project of the LF and “umbrella” for other standards workgroups
- Workgroups produce standards in specialized areas of the Linux platform space (a11y, i18n, printing, etc.)
- LSB and workgroups are aligned around a common roadmap (LSB roadmap)
- Workgroups produce deliverables (specs, tests, etc.) in a standard format (LSB module)
- Workgroups are expected to produce standards that are eventually broadly adopted
 - *Any activity which is not on track for broad adoption is explicitly out of scope*



Safeguarding the
Future of Linux
Through Standards

Project organization (cont.)

- New structure allows domain experts to easily create new workgroups with ready-made processes, governance, context, and a delivery vehicle into the major distros
 - I.e., domain experts can focus on their problem domains
- New structure also allows the LSB to be more proactive, less reactive
 - Forward looking work done in workgroups, not added to LSB till best practice
- Governance is very lightweight and modeled after IETF (see LSB Charter)



Safeguarding the
Future of Linux
Through Standards

LSB infrastructure project: Motivations

- Key LSB success factor is proper project infrastructure
 - How do we track the continually evolving open source landscape (independently evolving distros, upstream projects, etc.)?
 - Interface standard is only as good as its tests, so how do we maintain good testsuites?
- Example: inotify
 - Do all the major distros ship it? If not, when is it on their product roadmaps?
 - How do we communicate decisions?
 - What do we assert about inotify? How do we test it?



Safeguarding the
Future of Linux
Through Standards

LSB infrastructure project: Goals

- Improve test suite coverage (75% of LSB interfaces tested by LSB 4.0)
- Make test execution more user-friendly = make it easier to LSB certify distros
- Involve distros, ISVs, and upstreams more directly in the LSB development process = more coordinated roadmaps, economies of scale (import existing tests, export tests to upstreams so they can maintain them)
- Make it easier to evolve the LSB over time
- Make better decisions, make it easier to communicate decisions and roadmaps



Safeguarding the
Future of Linux
Through Standards

LSB infrastructure project: Methods

- *Specification*, which defines the requirements of LSB interfaces (at assertion granularity);
- *Upstream components* (e.g. glibc), which implement the LSB interfaces;
- *Distributions*, which are collections of upstream components at various points in time (and which define which interfaces are candidates for inclusion in the specification);
- *Tests* that check whether the assertions in the specification hold true;
- *Documentation* of the interfaces for ISVs;
- *Certification and regression testing*, to keep all the LSB implementations aligned



Safeguarding the
Future of Linux
Through Standards

More information

<http://www.linux-foundation.org/>
<http://developer.linux-foundation.org/>



Safeguarding the
Future of Linux
Through Standards

LSB 3.0

- LSB C++ provides first portable C++ standard for Linux platforms
- Certified distributions:
 - Asianux 2.0
 - Red Hat Enterprise Linux 4
 - SUSE Linux Enterprise Server 9



Safeguarding the
Future of Linux
Through Standards

LSB 3.1

- LSB Core becomes ISO standard (ISO/IEC 23360)
- LSB Desktop introduced
 - initially covers GUI toolkits (Gtk, Qt)
 - just the beginning of desktop support!
 - in short, it is now possible to use the LSB to write portable Linux desktop apps



Safeguarding the
Future of Linux
Through Standards

LSB 3.1 (cont.)

- All major distros committed to LSB 3.1 certification:
 - Asianux 2.0 SP1
 - Debian “etch”
 - Mandriva Corporate Server 4.0/Corporate Desktop 4.0
 - Red Hat Enterprise Linux 5
 - SUSE Linux Enterprise Server 10/Enterprise Desktop 10
 - Ubuntu 6.06 (“dapper”)
 - Many more



Safeguarding the
Future of Linux
Through Standards

LSB 3.2

- Perl
- Python
- Desktop: Four freedesktop.org specifications (Desktop base directory, Desktop entry, Icon theme, menu)
 - Portland?
- Printing: Driver support, application interface (CUPS)
- Multimedia (ALSA, libao)
- OpenI18n
- Accessibility



Safeguarding the
Future of Linux
Through Standards

LSB 3.2 (cont.)

- Release target: Q2 2007
- Goal: Maintain LSB certification for LSB 3.1 certified distros (Asianux 3.0, RHEL 5, SLE 10, Debian etch, Ubuntu dapper, etc.)



Safeguarding the
Future of Linux
Through Standards

LSB 4.0

- Binary compatibility with LSB 3
 - I.e., LSB 3 apps should run on LSB 4 certified distros without a recompile
- GCC 4.1
- glibc 2.4
- Packaging
 - ISVs want to use their own installation method (install script, InstallAnywhere)
- Java?
- OpenSSL
- ... You tell me!



Safeguarding the
Future of Linux
Through Standards

LSB 4.0 (cont.)

- Release target: 2008 (largely depends on RHEL 6, SLE 11, etc. release schedule)
- “Certified on day one” targets:
 - Asianux 3.0 (via Service Pack)
 - Debian “etch+1”
 - Ubuntu LTS “dapper+1”
 - RHEL 6
 - SLE 11
 - Fedora, Freespire/Linspire, Mandriva, Sun Wah, SUSE Linux, Xandros, ...



Safeguarding the
Future of Linux
Through Standards

LSB SDK, Developer Network

- LSB SDK bundles all tools developers need to build portable Linux apps
- LSB application testkit aims to be the LSB's `validator.w3.org`
- LSB Developer Network provides central source for developers writing portable Linux apps (documentation, tutorials, etc.) (“the MSDN for Linux”)
 - Bottom up vs. top down construction
- Goal: Make it “dead simple” to build apps for Linux that are portable across distros
- <http://developer.freestandards.org/>