



Safeguarding the
Future of Linux
Through Standards

LSB overview and progress report

Ian Murdock
Chief Technology Officer
Free Standards Group



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 Free Standards Group offers certification program around the LSB interfaces
 - Distributions
 - Applications



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?

What makes a good platform (cont.)?

- Developer friendly
 - Central source of information for developers (“Developer Network”)
 - Developer tools
 - Documentation
 - 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



Safeguarding the
Future of Linux
Through Standards

Platform thinking

- Goals and motivations
 - LSB = “highest common denominator” across a well defined subset of the Linux distros
 - Must be possible for end users and ISVs to translate between LSB and distro releases
- 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.

LSB roadmap



Safeguarding the
Future of Linux
Through Standards

- **LSB 3.x (2006-2008)**
 - 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”)
 - ...
- **LSB 4.x (2008-2010)**
 - Asianux 3.0
 - Debian “etch” + 1
 - Mandriva Corporate 5.0
 - Red Hat Enterprise Linux 6
 - SUSE Linux Enterprise 11
 - Ubuntu LTS “dapper” + 1
 - ...

Platform thinking (cont.)

- 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
- Direct involvement of upstream developers, ISVs and OSVs in LSB development process
 - LSB as “anticipating standard”
 - 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

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



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

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



Safeguarding the
Future of Linux
Through Standards

- Core additions/readditions based on ISV feedback
- Perl
- Python
- Desktop: Four freedesktop.org specifications (Desktop base directory, Desktop entry, Icon theme, menu), Qt4, FreeType, Xft, Xrender
- Printing: Driver support (CUPS, ijs, OPVP), application interface (CUPS), printer driver additions to FHS
- Accessibility (Keyboard I/O, AT-SPI (optional))
- Multimedia (ALSA, libao)
- OpenI18n



Safeguarding the
Future of Linux
Through Standards

LSB 3.2 (cont.)

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

LSB 4.0



Safeguarding the
Future of Linux
Through Standards

- 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)
- dbus
- 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/>

LSB infrastructure project: Motivations



Safeguarding the
Future of Linux
Through Standards

- 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 test suites?
- 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

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



Safeguarding the
Future of Linux
Through Standards

Organizational issues

- LSB is “umbrella” and delivery vehicle for standards developed by FSG workgroups
 - LSB produces “core” standard
 - Other workgroups produce standards in more specialized areas (accessibility, printing, etc.)
- All FSG workgroups aligned around LSB roadmap, timeline
- All FSG workgroups produce standards in common format
 - “LSB module” = specs and tests that fit into new LSB framework developed by ISP RAS



Safeguarding the
Future of Linux
Through Standards

Organizational issues (cont.)

- LSB modules can be required or optional
- “The LSB” = all required modules at time of release integrated into single LSB spec
- LSB spec accompanied by product “testkits”, one for each certification offering
 - LSB distribution certification -> LSB Distribution Testkit
 - LSB application certification -> LSB Application Testkit
- Optional modules published but no certification offered around them



Safeguarding the
Future of Linux
Through Standards

LSB 3.2 timeline

- December 2006: LSB f2f (roadmap and timeline finalized, tasks assigned, etc.); alphas of LSB Distribution Testkit and LSB Application Testkit
- April 2007: New LSB infrastructure in beta, LSB 3.2 freeze
 - specs and tests final but unintegrated
- May 2007: LSB 3.2 Release Candidate, public comment period
 - specs and tests integrated into single LSB spec
- June 2007: New LSB infrastructure in production, LSB 3.2 released



Safeguarding the
Future of Linux
Through Standards

LSB Distribution Testkit

- Single download for distribution vendors that wish to test for LSB compliance/LSB certify
- Composed of test packages delivered via LSB modules in a single tar file (much like today's `lsb-runtime-tests` bundle)
- `lsb-autotest` wraps test packages, provides single execution and test result file
- Test Execution Framework (TEF) adds web front end (via localhost web server), additional reporting/visualization capabilities, integration with FSG certification system



Safeguarding the
Future of Linux
Through Standards

LSB Application Testkit

- Single download for application vendors that wish to test for LSB compliance/LSB certify
- Aims to be the LSB's validator.w3.org
 - Extremely lightweight, ingrained in the developer consciousness
- Composed of new lsbappchk that supports new interactive mode
- Does not require root access to install and run
- Output contains hyperlinks into Developer Network so users can easily find additional information about error messages and what to do about them



Safeguarding the
Future of Linux
Through Standards

LSB SDK

- Current LSB SDK makes it easy to create an LSB compliant *binary* but not yet easy enough to create a distributable, easy to install LSB compliant *application*
- Longer term: Will discuss at Packaging Summit
- Shorter term:
 - Can we bundle RPM helpers, alien, APT/yum repo generators, etc. so LSB apps can be installed with distro's preferred package manager?
 - Can we include tools that make it easy to create self-contained apps, i.e., bundle shared objects to satisfy LSB requirements, etc.?