

Subject: Codeset Independent ldterm(7M) and stty(1)
Submitted by: Ienup Sung
File: PSARC/1999/140/opinion.ms
Date: July 21st, 1999
Committee: Joseph Kowalski, Tim Marsland, Terrence Miller, David Robinson, Glenn Skinner.
Steering Committee:
Solaris Operating Environment
soesc-opinion@scs.eng

Operating Systems and Networking
onsc@eng

1. Summary

The current ldterm(7M) and stty(1) implementations are EUC codeset specific and contain EUC representation dependencies. This project is to provide a codeset independent ldterm(7M) module and stty(1) command. Currently, non-EUC locales push a pair of code conversion modules around the ldterm module to circumvent this restriction. This procedure is quite cumbersome and error prone.

2. Decision & Precedence Information

The project is approved as specified in reference [1], but as modified by the required technical changes listed in Appendix A below.

The project may be delivered in a minor release of Solaris.

3. Interfaces

The project exports the following interfaces.

box; c s s l 1 1.	Interfaces	Exported	_ Interface	Classification	Comments	_
stty(1)	Stable	T{ defeuwc	behavior in non-EUC locales.	T} ldterm(7m)	Stable	T{
Expanded definition of EUC_WSET and EUC_WGET in non-EUC locales.	T} T{ ldterm(7m)	CSDATA_SET	CSDATE_GET	T} T{ Consolidation Private	T} T{ ioctls only used by stty(1)	
T} ldterm.dat	Sun Private	T{ Data file name and format as defined in [1].	T}			

4. Opinion

Originally this project was presented in a form considerably different than the final proposal. That design downloaded large amounts of data through multiple calls to new ldterm ioctls. The committee had two major issues with this implementation.

1. The switch between codesets was not atomic. The behavior was not well defined when a codeset width table was partially downloaded.
2. The large downloaded codeset width tables were per-stream and not shared. This could easily result in massive amounts of physical kernel memory being consumed.

The revised proposal addressed both of these concerns. It is worth noting that although the committee suggested several possible ways to address its concerns, the project team returned with an implementation which was not discussed. This is mentioned here because project teams have expressed concern about being constrained by solutions suggested by the committee. This should not be the case and wasn't with this project.

The committee was still concerned by the physical memory consumed by the project which is potentially 280 kilobytes but is only 16 kilobytes in the initial implementation. The additional memory will be consumed as additional Unicode code-planes are defined. After consideration, the committee decided that because the memory consumption was not extremely excessive and much of it was deferred in time the project was acceptable as proposed. However, this concern is captured as a Technical Change Advised. Finally, the committee was insistent that stty options continue to have the same semantics as before. This discussion resulted in the two Technical Changes Required for this project.

5. Minority Opinion(s)

None.

6. Advisory Information

None.

7. Appendices

7.1. Appendix A: Technical Changes Required

1. The project must support the existing “stty -g” behavior. Specifically, that is the ability to produce output in a form appropriate as an argument to another stty command. With this project, that should include the value of the non-EUC code set.
2. The output of ‘stty -a’ should report the “code set width data” name. This is usually the locale name, but the label “locale” should not be used in the stty output to allow a future project which may more correctly require the “locale” label. Differences could arise where sub-locales exist or character sets are shared among locales. The committee suggested (but did not require) the identifier “csdata” be used.

7.2. Appendix B: Technical Changes Advised

1. The project is advised to not lock down physical memory for a large Unicode width table when not required. Possible implementations include using pageable kernel memory or using an optionally loadable module.

7.3. Appendix C: Reference Material

Unless stated otherwise, path names are relative to the case directory PSARC/1999/140.

1. Project Specification (as if for a fast-track).
File: commit.materials/spec.
2. Technical Description
File: commit.materials/ldterm-csi.txt
3. Diff-marked man page.
File: commit.materials/stty.1
4. Diff-marked man page.
File: commit.materials/ldterm.7m
5. Project Plan
File: incept.materials/ldterm_csi_projplan.ps
6. One Pager
File: 990223_ienup.sung
7. Dependency Discussion.
File: incept.materials/stty-euc-dependency.txt
8. Dependency Discussion.
File: incept.materials/ldterm-euc-dependency.txt

9. Background Codeset Information.
File: `commit.materials/csicookbookappendix.pdf`
10. Solaris Internationalization Guide for Developers Overview of `en_US.UTF-8` Locale Support
File: User documentation (current).