Chez Scheme Version 7.4d Release Notes
Copyright © 2009 Cadence Research Systems
All Rights Reserved
July 2009

1. Overview

This document outlines the changes made to *Chez Scheme* for Version 7.4d since Version 7.4.

Version 7.4d is available for the following platforms:

- Linux x86, threaded and nonthreaded
- MacOS X x86, threaded and nonthreaded
- MacOS X PowerPC, threaded and nonthreaded
- Windows x86, threaded and nonthreaded
- OpenBSD x86, threaded and nonthreaded
- FreeBSD x86, threaded and nonthreaded
- Solaris 32-bit Sparc, threaded and nonthreaded
- Solaris 64-bit Sparc, threaded and nonthreaded

This document contains three sections describing significant (1) functionality changes, (2) bugs fixed, and (3) performance enhancements. A version number listed in parentheses in the header for a change indicates the first minor release or internal prerelease to support the change.

More information on *Chez Scheme* and *Petite Chez Scheme* can be found at http://www.scheme.com, and extensive documentation is available in *The Scheme Programming Language, 3rd edition* and the *Chez Scheme Version 7 User's Guide*.

2. Functionality Changes

2.1. Tarball configuration change (7.4c)

The configure script used when installing Chez Scheme from a tarball now forces the executable to be rebuilt to link against local libraries by default. Use the `--noforce-relink` to avoid rebuilding the executable.

2.2. Windows libraries (7.4b/7.4c)

Windows builds of *Chez Scheme* now link against the msvcr90.dll C run-time library. Manifests are incorporated directly into the Chez Scheme dll and exe files, which should allow Windows to find the C run-time library in the same directory as the Chez Scheme dll and/or exe file if the run-time library is not already installed on the system. For the threaded version, the pthread library also links against msvcr90.dll and has been renamed accordingly to avoid conflicts with versions built against different C run-time libraries.

Static libraries linked against the static libcmt are also available, as well as static libraries built to link against msvcr90.dll.

2.3. C library routines for 32- and 64-bit integers (7.4b)

Four new C library operators have been added for converting 32- and 64-bit integers from their Scheme representations:

- `Sinteger32_value(`x`)` returns the 32-bit integer value of the Scheme exact integer x;
- `Sunsigned32_value(`x`)` returns the 32-bit unsigned integer value of the Scheme exact integer x;
- `Sinteger64_value(`x`)` returns the 64-bit integer value of the Scheme exact integer x; and
- `Sunsigned64_value(`x`)` returns the 64-bit unsigned integer value of the Scheme exact integer x.

An error is signaled if a 32-bit value is not in the range -2^{31} through $2^{32}-1$ or a 64-bit values is not in the range -2^{63} through $2^{64}-1$.

Simiarly, four new C library operators have been added for converting 32- and 64-bit integers to their Scheme representations:

- `Sinteger32(`x`)`,
- `Sunsigned32(`x`)`,
- `Sinteger64(`x`)`, and
- `Sunsigned64(`x`)`.

Each returns a Scheme exact integer whose value is x. The arguments are signed or unsigned 32- or 64-bit C integers, as appropriate.

2.4. Months range from 1 to 12 (7.4b)

The month field of a date record, e.g., one returned by `current-date`, runs from 1 through 12 rather than 0 through 11, for compatibility with SRFI 19.

2.5. Monotonic time (7.4b)

Requesting monotonic no longer fails on Linux systems that do not support monotonic clocks; on such systems, the real-time clock is now used instead. All of the high-precision time procedures now fall back on older time mechanisms when requesting high-precision timers fails.

2.6. New and altered expression-editor key bindings (7.4b)

The expression editor now binds the common delete-key sequence (`Esc[3~`) to `ee-delete-char` and escape followed by the same sequence to `ee-delete-sexp`. The effect of this is that the delete key should now delete forward rather than backward. The backspace key should still delete backward.

New bindings have also been added for the home and end keys of certain terminal emulators, including the Gnome terminal.

2.7. Top-level value handling (7.4b)

The `top-level-value` and `top-level-bound?` procedures now recognize as bound a variable assigned by a file compiled in one session and loaded into another session—even if the variable was not defined.

3. Bug Fixes

3.1. `fresh-line` on unbuffered file ports (7.4d)

A bug in the handling of the port “beginning of line” flag on unbuffered output ports, resulting in missing or extra newlines being emitted by `fresh-line`, has been fixed.

3.2. Work-around for automatic margins on MacOS X Terminal (7.4d)

The expression editor disables automatic margins (i.e., automatic line wrapping) when possible so it can use the last column without scrolling the display. Unfortunately, when automatic margins are disabled in the MacOS X Terminal application, they are disabled simultaneously for all windonws, not just for the current window, causing problems when the expression editor is running in one window and long lines are printed in another. The expression editor now disables automatic margins only temporarily while writing text to the last column, reducing (though not entirely eliminating) the possibility that other windows are affected.

3.3. Invalid memory reference using hashtables (7.4c)

A bug that caused corruption of the heap and typically resulted in an “unrecoverable invalid memory reference” while using the R6RS eq hashtable interface has been fixed.

3.4. Expression editor datum comment handling (7.4b)

The expression editor no longer returns end-of-file for an entry containing only a single commented-out datum.

3.5. Expression editor clipboard bug under Windows (7.4b)

The expression editor now properly closes the clipboard after a paste (control-v) operation, i.e., no longer prevents new items from being copied to the clipboard.

3.6. Bug in `exact->inexact` (7.4b)

A bug that could cause `exact->inexact` to improperly round an exact value halfway between two floating-point values has been fixed.

3.7. Bug in `format` tabulation (7.4b)

A bug resulting in an “undefined for zero” error when the optional column width specifier in a tabulate directive is zero, e.g., `~1,0t`, has been fixed.

3.8. Bug in `ash` (7.4b)

A bug that caused `ash` to return the wrong result for word-sized or greater negative (right) shifts of large powers of two has been fixed.

3.9. Expansion internal error (7.4b)

A bug in the expander that sometimes resulted in an internal error ("source-wrap ae/x mismatch") has been fixed.

3.10. Bug in handling of `compile` (7.4b)

A bug in the compiler that caused it to treat the return value of `compile` as true in test context has been fixed.

4. Performance Enhancements

4.1. Standard (usually larger) file buffers (7.4b/7.4c)

`open-input-file`, `open-output-file`, and related procedures now use the default file buffer size as determined by the maximum of the stdio.h constant `BUFSIZ` and 4096, often resulting in increased throughput.

4.2. Improved register assignment for x86 (7.4b)

A better choice of register assignments has resulted in measurable improvement in x86 performance. Actual improvements differ from one program to another, and we've seen anything from -9% descreases in speed to 36% increases, with increases of 5-15% appearing typical.