The Scheme Requests for Implementation (a.k.a. SRFI) process allows individual members of the Scheme community to propose libraries and extensions to be supported by multiple Scheme implementations.

PLT Scheme is distributed with implementations of many SRFIs, most of which can be implemented as libraries. To import the bindings of SRFI n, use

```
(require srfi/n)
```

This document lists the SRFIs that are supported by PLT Scheme and provides a link to the original SRFI specification (which is also distributed as part of PLT Scheme’s documentation).
SRFI 1: List Library

(require srfi/1)

Original specification: [SRFI 1]

This SRFI works with pairs and lists as in scheme and mzscheme, which are immutable, so it does not export set-car! and set-cdr!. The other provided bindings that end in ! are equivalent to the corresponding bindings without !.
SRFI 2: AND-LET*: an AND with local bindings...

(require srfi/2)

Original specification: SRFI 2
SRFI 4: Homogeneous numeric vector datatypes

(require srfi/4)

Original specification: SRFI 4

This SRFI’s reader and printer syntax is not supported. The bindings are also available from scheme/foreign.
SRFI 5: A compatible let form with signatures and rest arguments

(require srfi/5)

Original specification: SRFI 5
SRFI 6: Basic String Ports

(require srfi/6)

Original specification: [SRFI 6]

This SRFI's bindings are also available in scheme/base.
SRFI 7: Feature-based program configuration language

(require srfi/7)

Original specification: SRFI 7
SRFI 8: RECEIVE: Binding to multiple values

(require srfi/8)

Original specification: SRFI 8
SRFI 9: Defining Record Types

(require srfi/9)

Original specification: SRFI 9
SRFI 11: Syntax for receiving multiple values

(require srfi/11)

Original specification: [SRFI 11]

This SRFI’s bindings are also available in scheme/base, but without support for dotted “rest” bindings.
SRFI 13: String Libraries

(require srfi/13)

Original specification: SRFI 13
SRFI 14: Character-set Library

(require srfi/14)

Original specification: SRFI 14
SRFI 16: Syntax for procedures of variable arity

(require srfi/16)

Original specification: [SRFI 16]

This SRFI's bindings are also available in scheme/base.
SRFI 17: Generalized set!

(require srfi/17)

Original specification: SRFI 17
SRFI 19: Time Data Types and Procedures

(require srfi/19)

Original specification: SRFI 19

Care most be taken NOT to confuse the internal date structure with the PLT Scheme date; they are not the same, and all procedures from the SRFI library expect the former.
SRFI 23: Error reporting mechanism

(require srfi/23)

Original specification: [SRFI 23]

This SRFI's bindings are also available in scheme/base.
SRFI 25: Multi-dimensional Array Primitives

(require srfi/25)

Original specification: SRFI 25
SRFI 26: Notation for Specializing Parameters without Currying

(require srfi/26)

Original specification: [SRFI 26]
SRFI 27: Sources of Random Bits

(require srfi/27)

Original specification: SRFI 27
SRFI 28: Basic Format Strings

(require srfi/28)

Original specification: [SRFI 28]

This SRFI's bindings are also available in scheme/base.
SRFI 29: Localization

(require srfi/29)

Original specification: SRFI 29
SRFI 30: Nested Multi-line Comments

(require srfi/30)

Original specification: SRFI 30

This SRFI’s syntax is part of PLT Scheme’s default reader.
SRFI 31: A special form rec for recursive evaluation

(require srfi/31)

Original specification: SRFI 31
SRFI 34: Exception Handling for Programs

(require srfi/34)

Original specification: SRFI 34
SRFI 35: Conditions

(require srfi/35)

Original specification: SRFI 35
SRFI 38: External Representation for Data With Shared Structure

(require srfi/38)

Original specification: [SRFI 38]

This SRFI’s syntax is part of PLT Scheme’s default reader and printer.
SRFI 39: Parameter objects

(require srfi/39)

Original specification: SRFI 39

This SRFI's bindings are also available in scheme/base.
SRFI 40: A Library of Streams

(require srfi/40)

Original specification: SRFI 40

Superceded by srfi/41.
SRFI 41: Streams

(require srfi/41)

Original specification: SRFI 41
SRFI 42: Eager Comprehensions

(require srfi/42)

Original specification: [SRFI 42](#)

Forms that syntactically detect if recognize both if from `scheme/base` and if from `mzscheme`.
SRFI 43: Vector Library

(requiresrfi/43)

Original specification: SRFI 43
SRFI 45: Primitives for Expressing Iterative Lazy Algorithms

(require srfi/45)

Original specification: [SRFI 45]

Additional binding:

promise? v → boolean?
  v : any/c

Returns #t if v is a promise, #f otherwise.
SRFI 48: Intermediate Format Strings

(reuse srfi/48)

Original specification: SRFI 48
SRFI 54: Formatting

(require srfi/54)

Original specification: SRFI 54
SRFI 57: Records

(require srfi/57)

Original specification: SRFI 57
SRFI 59: Vicinity

(require srfi/59)

Original specification: SRFI 59
SRFI 60: Integers as Bits

(requi re sr fi/ 60)

Original specification: SRFI 60
SRFI 61: A more general cond clause

(require srfi/61)

Original specification: SRFI 61
SRFI 62: S-expression comments

Original specification: SRFI 62

This SRFI’s syntax is part of PLT Scheme’s default reader (no require is needed).
SRFI 63: Homogeneous and Heterogeneous Arrays

(require srfi/63)

Original specification: SRFI 63
SRFI 64: A Scheme API for test suites

(require srfi/64)

Original specification: SRFI 64
SRFI 66: Octet Vectors

(require srfi/66)

Original specification: SRFI 66
SRFI 67: Compare Procedures

(require srfi/67)

Original specification: SRFI 67
SRFI 69: Basic hash tables

(require srfi/69)

Original specification: SRFI 69
SRFI 71: Extended LET-syntax for multiple values

(require srfi/71)

Original specification: SRFI 71
SRFI 74: Octet-Addressed Binary Blocks

(require srfi/74)

Original specification: SRFI 74
SRFI 78: Lightweight testing

(require srfi/78)

Original specification: SRFI 78
SRFI 86: MU & NU simulating VALUES & CALL-WITH-VALUES...

(require srfi/86)

Original specification: [SRFI 86]
SRFI 87: => in case clauses

(require srfi/87)

Original specification: SRFI 87
SRFI 98: An interface to access environment variables

(require srfi/98)

Original specification: SRFI 98
SRFI 23: Error reporting mechanism, 17
SRFI 25: Multi-dimensional Array Primitives, 18
SRFI 26: Notation for Specializing Parameters without Currying, 19
SRFI 27: Sources of Random Bits, 20
SRFI 28: Basic Format Strings, 21
SRFI 29: Localization, 22
SRFI 2: AND-LET*: an AND with local bindings..., 4
SRFI 30: Nested Multi-line Comments, 23
SRFI 31: A special form rec for recursive evaluation, 24
SRFI 34: Exception Handling for Programs, 25
SRFI 35: Conditions, 26
SRFI 38: External Representation for Data With Shared Structure, 27
SRFI 39: Parameter objects, 28
SRFI 40: A Library of Streams, 29
SRFI 41: Streams, 30
SRFI 42: Eager Comprehensions, 31
SRFI 43: Vector Library, 32
SRFI 45: Primitives for Expressing Iterative Lazy Algorithms, 33
SRFI 48: Intermediate Format Strings, 34
SRFI 4: Homogeneous numeric vector datatypes, 5
SRFI 54: Formatting, 35
SRFI 57: Records, 36
SRFI 59: Vicinity, 37
SRFI 5: A compatible let form with signatures and rest arguments, 6
SRFI 60: Integers as Bits, 38
SRFI 61: A more general cond clause, 39
SRFI 62: S-expression comments, 40
SRFI 63: Homogeneous and Heterogeneous Arrays, 41
SRFI 64: A Scheme API for test suites, 42
SRFI 66: Octet Vectors, 43
SRFI 67: Compare Procedures, 44
SRFI 69: Basic hash tables, 45
SRFI 6: Basic String Ports, 7
SRFI 71: Extended LET-syntax for multiple values, 46
SRFI 74: Octet-Addressed Binary Blocks, 47
SRFI 78: Lightweight testing, 48
SRFI 7: Feature-based program configuration language, 8
SRFI 86: MU & NU simulating VALUES & CALL-WITH-VALUES..., 49
SRFI 87: => in case clauses, 50
SRFI 8: RECEIVE: Binding to multiple values, 57
SRFI 98: An interface to access environment variables, 51
SRFI 9: Defining Record Types, 10
srfi/1, 3
srfi/11, 11
srfi/13, 12
srfi/14, 13
srfi/16, 14
srfi/17, 15
srfi/19, 16
srfi/2, 4
srfi/23, 17
srfi/25, 18
srfi/26, 19
srfi/27, 20
srfi/28, 21
srfi/29, 22
srfi/30, 23
srfi/31, 24
srfi/34, 25
srfi/35, 26
srfi/38, 27
srfi/39, 28
srfi/4, 5
srfi/40, 29
srfi/41, 30
srfi/42, 31
srfi/43, 32
srfi/45, 33
srfi/48, 34
srfi/5, 6
vector-swap!, 32
vector-unfold, 32
vector-unfold-right, 32
vector=, 32
vector?, 32
vicinity:suffix?, 37
with-exception-handler, 25
write-with-shared-structure, 27
xcons, 3
xsubstring, 12
zip, 3