

# s11n egg

---

Serialization and deserialization of arbitrary objects  
Extension for Chicken Scheme  
Version 0.6

## Table of Contents

<b>1</b>	<b>About this egg .....</b>	<b>1</b>
1.1	Version history .....	1
1.2	Usage .....	1
<b>2</b>	<b>Documentation .....</b>	<b>2</b>
<b>3</b>	<b>License .....</b>	<b>3</b>
	<b>Index .....</b>	<b>4</b>

# 1 About this egg

## 1.1 Version history

- 0.6        Nasty little bug in backreferences fixed
- 0.5        Hash-table-deserialization bug reported by Kabir Soorya
- 0.4        Fixed small bug in port-serialization; added note about continuations
- 0.3        Serialization of hash-tables is more storage efficient; added ‘chicken-dump’ program
- 0.2        Failure handling for some cases was invalid
- 0.1        Initial release

## 1.2 Usage

Load this egg like so:

```
(require-extension s11n)
```

## 2 Documentation

This extension allows serializing and deserializing arbitrary data (including procedures and continuations) into/from ports. Circular data is support as well as uninterned symbols. Foreign-pointer objects and ports other than the default input-, output- and error-ports can not be serialized. Threads may be serialized provided they are not running or ready (this means suspended or created but not yet started threads).

The serialized data is endianness- and word-size dependent.

To enable serialization of procedures and continuations, CHICKEN has to be built with the `--enable-procedure-tables` configuration option. In particular, any compiled module that is referenced in serialized data has to be compiled with a CHICKEN version that has procedure tables enabled.

Compiled procedures and continuations can be deserialized, provided that the same executable or libraries are loaded/linked - the compiled code must be available so that the deserialization process can find the associated code.

Continuations can be serialized but care has to be taken about what exactly is stored in such an object. Every invocation of `call-with-current-continuation` includes the complete list of pending `dynamic-wind` thunks (and their dynamic environment) which is likely to include non-serializable state.

Since the deserialization process breaks any assumptions about a unique identity of objects, TinyCLOS instances and classes can currently not be deserialized.

CHICKEN version 2.207 or higher is required to use this extension.

**serialize** [procedure]  
 (serialize X [PORT [SERIALIZER]])

Writes a binary representation of X into PORT which defaults to the value of (current-output-port). If the serialization is unable to handle some object, the procedure `SERIALIZER` is invoked with that object as it's sole argument. The procedure should either signal an error or return a placeholder object that is to be serialized instead. If no serializer is specified, an error will be signalled.

**deserialize** [procedure]  
 (deserialize [PORT])

Reads a binary representation of a Scheme data object from PORT which defaults to the value of (current-input-port). The deserialized object is returned.

A separate tool is also available, named `chicken-dump` that writes a description about the contents of a serialized file. Just invoke

**chicken-dump** FILENAME  
 on the command prompt.

### 3 License

Copyright (c) 2005, Felix Winkelmann. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ASIS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Index

D

deserialize ..... 2

S

serialize..... 2