



Open Source Used In Polaris UC Apps RM 17.10.1

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1329598769

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please contact us at external-opensource-requests@cisco.com.

In your requests please include the following reference number 78EE117C99-1329598769

Contents

1.1 libsrtp 2.0.0

1.1.1 Available under license

1.2 parson 1.0

1.2.1 Available under license

1.1 libsrtp 2.0.0

1.1.1 Available under license :

```
/*
 *
 * Copyright (c) 2001-2006 Cisco Systems, Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * Redistributions in binary form must reproduce the above
 * copyright notice, this list of conditions and the following
 * disclaimer in the documentation and/or other materials provided
 * with the distribution.
 *
 * Neither the name of the Cisco Systems, Inc. nor the names of its
 * contributors may be used to endorse or promote products derived
 * from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
```

* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
* INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

1.2 parson 1.0

1.2.1 Available under license :

/*

Parson (<http://kgabis.github.com/parson/>)

Copyright (c) 2012 Krzysztof Gabis

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 "AS IS", 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.

*/

```
#ifndef parson_parson_h
```

```
#define parson_parson_h
```

```
#ifdef __cplusplus
```

```
extern "C"
```

```
{
```

```
#endif
```

```

#include <stddef.h> /* size_t */

/* Types and enums */
typedef struct json_object_t JSON_Object;
typedef struct json_array_t JSON_Array;
typedef struct json_value_t JSON_Value;

typedef enum json_value_type {
    JSONError = 0,
    JSONNull = 1,
    JSONString = 2,
    JSONNumber = 3,
    JSONObject = 4,
    JSONArray = 5,
    JSONBoolean = 6
} JSON_Value_Type;

/* Parses first JSON value in a file, returns NULL in case of error */
JSON_Value * json_parse_file(const char *filename);

/* Parses first JSON value in a string, returns NULL in case of error */
JSON_Value * json_parse_string(const char *string);

/* JSON Object */
JSON_Value * json_object_get_value (const JSON_Object *object, const char *name);
const char * json_object_get_string (const JSON_Object *object, const char *name);
JSON_Object * json_object_get_object (const JSON_Object *object, const char *name);
JSON_Array * json_object_get_array (const JSON_Object *object, const char *name);
double json_object_get_number (const JSON_Object *object, const char *name);
int json_object_get_boolean(const JSON_Object *object, const char *name);

/* dotget functions enable addressing values with dot notation in nested objects,
just like in structs or c++/java/c# objects (e.g. objectA.objectB.value).
Because valid names in JSON can contain dots, some values may be inaccessible
this way. */
JSON_Value * json_object_dotget_value (const JSON_Object *object, const char *name);
const char * json_object_dotget_string (const JSON_Object *object, const char *name);
JSON_Object * json_object_dotget_object (const JSON_Object *object, const char *name);
JSON_Array * json_object_dotget_array (const JSON_Object *object, const char *name);
double json_object_dotget_number (const JSON_Object *object, const char *name);
int json_object_dotget_boolean(const JSON_Object *object, const char *name);

/* Functions to get available names */
size_t json_object_get_count(const JSON_Object *object);
const char * json_object_get_name (const JSON_Object *object, size_t index);

/* JSON Array */
JSON_Value * json_array_get_value (const JSON_Array *array, size_t index);

```

```
const char * json_array_get_string (const JSON_Array *array, size_t index);
JSON_Object * json_array_get_object (const JSON_Array *array, size_t index);
JSON_Array * json_array_get_array (const JSON_Array *array, size_t index);
double    json_array_get_number (const JSON_Array *array, size_t index);
int       json_array_get_boolean(const JSON_Array *array, size_t index);
size_t    json_array_get_count (const JSON_Array *array);
```

```
/* JSON Value */
```

```
JSON_Value_Type json_value_get_type (const JSON_Value *value);
JSON_Object * json_value_get_object (const JSON_Value *value);
JSON_Array * json_value_get_array (const JSON_Value *value);
const char * json_value_get_string (const JSON_Value *value);
double    json_value_get_number (const JSON_Value *value);
int       json_value_get_boolean(const JSON_Value *value);
void      json_value_free (JSON_Value *value);
```

```
#ifdef __cplusplus
```

```
}
```

```
#endif
```

```
#endif
```

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2022 Cisco Systems, Inc. All rights reserved.