

**NAME**

OSSP::uuid – OSSP **uuid** Perl Binding

**DESCRIPTION**

**OSSP uuid** is a ISO-C:1999 application programming interface (API) and corresponding command line interface (CLI) for the generation of DCE 1.1, ISO/IEC 11578:1996 and RFC 4122 compliant *Universally Unique Identifier* (UUID). It supports DCE 1.1 variant UUIDs of version 1 (time and node based), version 3 (name based, MD5), version 4 (random number based) and version 5 (name based, SHA-1). Additional API bindings are provided for the languages ISO-C++:1998, Perl:5 and PHP:4/5. Optional backward compatibility exists for the ISO-C DCE-1.1 and Perl Data::UUID APIs.

**OSSP::uuid** is the Perl binding to the **OSSP uuid** API. Three variants are provided:

**TIE-STYLE API**

The TIE-style API is a functionality-reduced wrapper around the OO-style API and intended for very high-level convenience programming:

```
use OSSP::uuid;
tie my $uuid, 'OSSP::uuid::tie', $mode, ...;
$uuid = [ $mode, ... ];
print ``UUID=$uuid\n``;
untie $uuid;
```

**OO-STYLE API**

The OO-style API is a wrapper around the C-style API and intended for high-level regular programming.

```
use OSSP::uuid;
my $uuid = new OSSP::uuid;
$uuid->load($name);
$uuid->make($mode, ...);
$result = $uuid->isnil();
$result = $uuid->compare($uuid2);
$uuid->import($fmt, $data_ptr);
$data_ptr = $uuid->export($fmt);
[!$str[, $rc]] = $uuid->error();
$ver = $uuid->version();
undef $uuid;
```

Additionally, the strings "v1", "v3", "v4", "v5" and "mc" can be used in \$mode and the strings "bin", "str", and "txt" can be used for \$fmt.

**C-STYLE API**

The C-style API is a direct mapping of the **OSSP uuid** ISO-C API to Perl and is intended for low-level programming. See *uuid(3)* for a description of the functions and their expected arguments.

```
use OSSP::uuid qw(:all);
my $uuid; $rc = uuid_create($uuid);
$rc = uuid_load($uuid, $name);
$rc = uuid_make($uuid, $mode, ...);
$rc = uuid_isnil($uuid, $result);
$rc = uuid_compare($uuid, $uuid2, $result);
$rc = uuid_import($uuid, $fmt, $data_ptr, $data_len);
$rc = uuid_export($uuid, $fmt, $data_ptr, $data_len);
$str = uuid_error($rc);
$ver = uuid_version();
$rc = uuid_destroy($uuid);
```

Additionally, the following constants are exported for use in \$rc, \$mode, \$fmt and \$ver:

```
UUID_VERSION,  UUID_LEN_BIN,  UUID_LEN_STR,  UUID_RC_OK,  UUID_RC_ARG,
UUID_RC_MEM,   UUID_RC_SYS,  UUID_RC_INT,  UUID_RC_IMP,  UUID_MAKE_V1,
UUID_MAKE_V3,  UUID_MAKE_V4,  UUID_MAKE_V5,  UUID_MAKE_MC,  UUID_FMT_BIN,
UUID_FMT_STR,  UUID_FMT_SIV,  UUID_FMT_TXT.
```



**EXAMPLES**

The following two examples create the version 3 UUID 02d9e6d5-9467-382e-8f9b-9300a64ac3cd, both via the OO-style and the C-style API. Error handling is omitted here for easier reading, but has to be added for production-quality code.

```
# TIE-style API (very high-level)
use OSSP::uuid;
tie my $uuid, 'OSSP::uuid::tie';
$uuid = [ "v1" ];
print "UUIDs: $uuid, $uuid, $uuid\n";
$uuid = [ "v3", "ns:URL", "http://www.ossps.org/" ];
print "UUIDs: $uuid, $uuid, $uuid\n";
untie $uuid;

# OO-style API (high-level)
use OSSP::uuid;
my $uuid = new OSSP::uuid;
my $uuid_ns = new OSSP::uuid;
$uuid_ns->load("ns:URL");
$uuid->make("v3", $uuid_ns, "http://www.ossps.org/");
undef $uuid_ns;
my $str = $uuid->export("str");
undef $uuid;
print "$str\n";

# C-style API (low-level)
use OSSP::uuid qw(:all);
my $uuid; uuid_create($uuid);
my $uuid_ns; uuid_create($uuid_ns);
uuid_load($uuid_ns, "ns:URL");
uuid_make($uuid, UUID_MAKE_V3, $uuid_ns, "http://www.ossps.org/");
uuid_destroy($uuid_ns);
my $str; uuid_export($uuid, UUID_FMT_STR, $str, undef);
uuid_destroy($uuid);
print "$str\n";
```

**SEE ALSO**

*uuid(1)*, *uuid-config(1)*, *uuid(3)*.

**HISTORY**

The Perl binding **OSSP::uuid** to **OSSP uuid** was implemented in November 2004 by Ralf S. Engelschall <rse AT engelschall DOT com>.

