NAME

Locale::Country - ISO codes for country identification (ISO 3166)

SYNOPSIS

```
use Locale::Country;

$country = code2country('jp');        # $country gets 'Japan'
$code    = country2code('Norway');    # $code gets 'no'

@codes   = all_country_codes();
@names   = all_country_names();

# semi-private routines
Locale::Country::alias_code('uk' => 'gb');
Locale::Country::rename_country('gb' => 'Great Britain');
```

DESCRIPTION

The `Locale::Country` module provides access to the ISO codes for identifying countries, as defined in ISO 3166-1. You can either access the codes via the *conversion routines* (described below), or with the two functions which return lists of all country codes or all country names.

There are three different code sets you can use for identifying countries:

alpha-2

Two letter codes, such as 'tv' for Tuvalu. This code set is identified with the symbol `LOCALE_CODE_ALPHA_2`.

alpha-3

Three letter codes, such as 'brb' for Barbados. This code set is identified with the symbol `LOCALE_CODE_ALPHA_3`.

numeric

Numeric codes, such as 064 for Bhutan. This code set is identified with the symbol `LOCALE_CODE_NUMERIC`.

All of the routines take an optional additional argument which specifies the code set to use. If not specified, it defaults to the two-letter codes. This is partly for backwards compatibility (previous versions of this module only supported the alpha-2 codes), and partly because they are the most widely used codes.

The alpha-2 and alpha-3 codes are not case-dependent, so you can use 'BO', 'Bo', 'bO' or 'bo' for Bolivia. When a code is returned by one of the functions in this module, it will always be lower-case.

As of version 2.00, Locale::Country supports variant names for countries. So, for example, the country code for "United States" is "us", so country2code('United States') returns 'us'. Now the following will also return 'us':

```
country2code('United States of America')
country2code('USA')
```

CONVERSION ROUTINES

There are three conversion routines: `code2country()`, `country2code()`, and `country_code2code()`.

code2country(CODE, [CODESET])

This function takes a country code and returns a string which contains the name of the country identified. If the code is not a valid country code, as defined by ISO 3166, then `undef` will be returned:

```
$country = code2country('fi');
```

country2code(STRING, [CODESET])

This function takes a country name and returns the corresponding country code, if such exists. If the argument could not be identified as a country name, then `undef` will be returned:

```
$code = country2code('Norway', LOCALE_CODE_ALPHA_3);
# $code will now be 'nor'
```

The case of the country name is not important. See the section *KNOWN BUGS AND LIMITATIONS* below.

country_code2code(CODE, CODESET, CODESET)

This function takes a country code from one code set, and returns the corresponding code from another code set.

```
$alpha2 = country_code2code('fin',
LOCALE_CODE_ALPHA_3, LOCALE_CODE_ALPHA_2);
# $alpha2 will now be 'fi'
```

If the code passed is not a valid country code in the first code set, or if there isn't a code for the corresponding country in the second code set, then `undef` will be returned.

QUERY ROUTINES

There are two function which can be used to obtain a list of all codes, or all country names:

`all_country_codes( [ CODESET ] )`

Returns a list of all two-letter country codes. The codes are guaranteed to be all lower-case, and not in any particular order.

`all_country_names( [ CODESET ] )`

Returns a list of all country names for which there is a corresponding country code in the specified code set. The names are capitalised, and not returned in any particular order.

Not all countries have alpha-3 and numeric codes - some just have an alpha-2 code, so you'll get a different number of countries depending on which code set you specify.

SEMI-PRIVATE ROUTINES

Locale::Country provides two semi-private routines for modifying the internal data. Given their status, they aren't exported by default, and so need to be called by prefixing the function name with the package name.

alias_code

Define a new code as an alias for an existing code:

```
Locale::Country::alias_code( ALIAS => CODE [, CODESET ] )
```

This feature was added as a mechanism for handling a "uk" code. The ISO standard says that the two-letter code for "United Kingdom" is "gb", whereas domain names are all .uk.

By default the module does not understand "uk", since it is implementing an ISO standard. If you would like 'uk' to work as the two-letter code for United Kingdom, use the following:

```
Locale::Country::alias_code('uk' => 'gb');
```

With this code, both "uk" and "gb" are valid codes for United Kingdom, with the reverse lookup returning "uk" rather than the usual "gb".

Note: this function was previously called _alias_code, but the leading underscore has been dropped. The old name will be supported for all 2.X releases for backwards compatibility.

rename_country

If the official country name just isn't good enough for you, you can rename a country. For example, the official country name for code 'gb' is 'United Kingdom'. If you want to change that, you might call:

```
Locale::Country::rename_country('gb' => 'Great Britain');
```

This means that calling code2country('gb') will now return 'Great Britain' instead of 'United Kingdom'. The original country name is retained as an alias, so for the above example, country2code('United Kingdom') will still return 'gb'.

EXAMPLES

The following example illustrates use of the `code2country()` function. The user is prompted for a country code, and then told the corresponding country name:

```
$| = 1;   # turn off buffering

print "Enter country code: ";
chop($code = <STDIN>);
$country = code2country($code, LOCALE_CODE_ALPHA_2);
if (defined $country)
{
    print "$code = $country\n";
}
else
{
    print "'$code' is not a valid country code!\n";
}
```

DOMAIN NAMES

Most top-level domain names are based on these codes, but there are certain codes which aren't. If you are using this module to identify country from hostname, your best bet is to preprocess the country code.

For example, **edu**, **com**, **gov** and friends would map to **us**; **uk** would map to **gb**. Any others?

KNOWN BUGS AND LIMITATIONS

- When using `country2code()`, the country name must currently appear exactly as it does in the source of the module. The module now supports a small number of variants.

  Possible extensions to this are: an interface for getting at the list of variant names, and regular expression matches.

- In the current implementation, all data is read in when the module is loaded, and then held in memory. A lazy implementation would be more memory friendly.

- Support for country names in different languages.

SEE ALSO

Locale::Language

ISO two letter codes for identification of language (ISO 639).

Locale::Script

ISO codes for identification of scripts (ISO 15924).

Locale::Currency

ISO three letter codes for identification of currencies and funds (ISO 4217).

Locale::SubCountry

ISO codes for country sub-divisions (states, counties, provinces, etc), as defined in ISO 3166-2. This module is not part of the Locale-Codes distribution, but is available from CPAN in CPAN/modules/by-module/Locale/

ISO 3166-1

The ISO standard which defines these codes.

http://www.iso.org/iso/en/prods-services/iso3166ma/index.html

Official home page for the ISO 3166 maintenance agency.

http://www.egt.ie/standards/iso3166/iso3166-1-en.html

Another useful, but not official, home page.

http://www.cia.gov/cia/publications/factbook/docs/app-d-1.html

An appendix in the CIA world fact book which lists country codes as defined by ISO 3166, FIPS 10-4, and internet domain names.

AUTHOR

Neil Bowers <neil@bowers.com>

COPYRIGHT