NAME

Locale::Codes - a distribution of modules to handle locale codes

DESCRIPTION

Locale-Codes is a distribution containing a set of modules. The modules each deal with different types of codes which identify parts of the locale including languages, countries, currency, etc.

Currently, the following modules are included:

Locale::Codes::Country, Locale::Country

This includes support for country codes (such as those listed in ISO-3166) to specify the country.

Because this module was originally distributed as Locale::Country, it is also available under that name.

Locale::Codes::Language, Locale::Language

This includes support for language codes (such as those listed in ISO-639) to specify the language.

Because this module was originally distributed as Locale::Language, it is also available under that name.

Locale::Codes::Currency, Locale::Currency

This includes support for currency codes (such as those listed in ISO-4217) to specify the currency.

Because this module was originally distributed as Locale::Currency, it is also available under that name.

Locale::Codes::Script, Locale::Script

This includes support for script codes (such as those listed in ISO-15924) to specify the script.

Because this module was originally distributed as Locale::Script, it is also available under that name.

Locale::Codes::LangExt

This includes support for language extension codes (such as those listed in the IANA language registry) to specify the language extension.

Locale::Codes::LangVar

This includes support for language variation codes (such as those listed in the IANA language registry) to specify the language variation.

Locale::Codes::LangFam

This includes support for language family codes (such as those listed in ISO 639-5) to specify families of languages.

Each module can support an arbitrary number of code sets, and it is not required that the relationship between these code sets be one-to-one. For example, the Locale::Codes::Country module supports code sets from ISO-3166 and the IANA standard, and they do not break the world down into exactly the same sets of countries. This does not cause any problem (though converting codes from ISO-3166 to IANA or back will not work except for countries that are one-to-one).

All data in all of these modules comes directly from the original standards (or as close to direct as possible), so it should be up-to-date at the time of release.

I plan on releasing a new version several times a year to incorporate any changes made in the standards. However, I don't always know about changes that occur, so if any of the standards change, and you want a new release sooner, just email me and I'll get one out.

In addition to the modules above, there are a number of support modules included in the distribution including:

      Locale::Codes
      Locale::Codes::Constants
      Locale::Codes::Country_codes
      Locale::Codes::Language_codes
      Locale::Codes::Currency_codes
      Locale::Codes::Script_codes
      Locale::Codes::LangExt_codes
      Locale::Codes::LangVar_codes
      Locale::Codes::LangFam_codes

These modules are not intended to be used by programmers. They contain functions or data that are used by the modules listed above.

NEW CODE SETS

I'm always open to suggestions for new code sets.

In order for me to add a code set, I want the following criteria to be met:

General-use code set

If a code set is not general use, I'm not likely to spend the time to add and support it.

An official source of data

I require an official (or at least, a NEARLY official) source where I can get the data on a regular basis.

Ideally, I'd only get data from an official source, but sometimes that is not possible. For example the ISO standards are not typically available for free, so I may have to get some of that data from alternate sources that I'm confident are getting their data from the official source. However, I will always be hesitant to accept a non-official source.

As an example, I used to get some country data from the CIA World Factbook. Given the nature of the source, I'm sure they're updating data from the official sources and I consider it "nearly" official. However, even in this case, I found that they were adding codes that were not part of the standard, so I have stopped using them as a source.

There are many 3rd party sites which maintain lists (many of which are actually in a more convenient form than the official sites). Unfortunately, I will reject most of them since I have no feel for how "official" they are.

A free source of the data

Obviously, the data must be free-of-charge. I'm not interested in paying for the data (and I'm not interested in the overhead of having someone else pay for the data for me).

A reliable source of data

The source of data must come from a source that I can reasonably expect to exist for the foreseeable future since I will be extremely reluctant to drop support for a data set once it's included.

I am also reluctant to accept data sent to me by an individual. Although I appreciate the offer, it is simply not practical to consider an individual contribution as a reliable source of data. The source should be an official agency of some sort.

These requirements are open to discussion. If you have a code set you'd like to see added, but which may not meet all of the above requirements, feel free to email me and we'll discuss it. Depending on circumstances, I may be willing to waive some of these criteria.

COMMON ALIASES

As of version 2.00, the modules supported common variants of names.

For example, Locale::Country supports variant names for countries, and a few of the most common ones are included in the data. The country code for "United States" is "us", so:

```
country2code('United States');
=> "us"
```

Now the following will also return 'us':

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

Any number of common aliases may be included in the data, in addition to the names that come directly from the standards. If you have a common alias for a country, language, or any other of the types of codes, let me know and I'll add it, with some restrictions.

For example, the country name ''North Korea'' never appeared in any of the official sources (instead, it was ''Korea, North'' or ''Korea, Democratic People's Republic of''. I would honor a request to add an alias ''North Korea'' since that's a very common way to specify the country (please don't request this... I've already added it).

On the other hand, a request to add Zaire as an alias for ''Congo, The Democratic Republic of'' will not be honored. The country's official name is no longer Zaire, so adding it as an alias violates the standard. Zaire was kept as an alias in versions prior to 3.00, but it has been removed. Other aliases (if any) which no longer appear in any standard (and which are not common variations of the name in the standards) have also been removed.

DEPRECATED CODES

Occasionally, a code is deprecated, but it may still be desirable to have access to it.

Although there is no way to see every code that has ever existed and been deprecated (since most codesets do not have that information available), as of version 3.20, every code which has ever been included in these modules can be referenced.

For more information, refer to the documentation on the code2XXX, XXX2code, all_XXX_codes, and all_XXX_names function in the **Locale::Codes::API** documentation.

SEE ALSO

Locale::Codes::API
　　The list of functions available in each of the modules listed below. The APIs for each module are exactly identical.

Locale::Codes::Country
　　Codes for identification of countries.

Locale::Codes::Language
　　Codes for identification of languages.

Locale::Codes::Script
　　Codes for identification of scripts.

Locale::Codes::Currency
　　Codes for identification of currencies and funds.

Locale::Codes::LangExt
　　Codes for identification of language extensions.

Locale::Codes::LangVar
　　Codes for identification of language variations.

Locale::Codes::LangFam
　　Codes for identification of language families.

Locale::Codes::Changes
　　A history of changes made to this distribution.

AUTHOR

Locale::Country and Locale::Language were originally written by Neil Bowers at the Canon Research Centre Europe (CRE). They maintained the distribution from 1997 to 2001.

Locale::Currency was originally written by Michael Hennecke and was modified by Neil Bowers for inclusion in the distribution.

From 2001 to 2004, maintenance was continued by Neil Bowers. He modified Locale::Currency for inclusion in the distribution. He also added Locale::Constants and Locale::Script.

From 2004-2009, the module was unmaintained.

In 2010, maintenance was taken over by Sullivan Beck (sbeck@cpan.org) with Neil Bower's permission. All problems or comments should be sent there. Alternately, problems can be reported using the perl problem tracker at:

> https://rt.cpan.org/Dist/Display.html?Queue=Locale-Codes

COPYRIGHT

```
Copyright (c) 1997-2001 Canon Research Centre Europe (CRE).
Copyright (c) 2001 Michael Hennecke (Locale::Currency)
Copyright (c) 2001-2010 Neil Bowers
Copyright (c) 2010-2014 Sullivan Beck
```

This module is free software; you can redistribute it and/or modify it under the same terms as Perl itself.