

COLLABORATORS			
TITLE :  GNU Libidn API Reference Manual			
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		March 2, 2015	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# **Contents**

1	GNU	U Libidn API Reference Manual	1
	1.1	idna	2
	1.2	stringprep	14
	1.3	punycode	25
	1.4	pr29	31
	1.5	tld	36
	1.6	idn-free	36
2	Inde	oy.	37

# **List of Figures**

1.1	Components of Libidn	 2

## **Chapter 1**

## **GNU Libidn API Reference Manual**

GNU Libidn is a fully documented implementation of the Stringprep, Punycode and IDNA specifications. Libidn's purpose is to encode and decode internationalized domain name strings. There are native C, C# and Java libraries.

The C library contains a generic Stringprep implementation. Profiles for Nameprep, iSCSI, SASL, XMPP and Kerberos V5 are included. Punycode and ASCII Compatible Encoding (ACE) via IDNA are supported. A mechanism to define Top-Level Domain (TLD) specific validation tables, and to compare strings against those tables, is included. Default tables for some TLDs are also included.

The Stringprep API consists of two main functions, one for converting data from the system's native representation into UTF-8, and one function to perform the Stringprep processing. Adding a new Stringprep profile for your application within the API is straightforward. The Punycode API consists of one encoding function and one decoding function. The IDNA API consists of the ToASCII and ToUnicode functions, as well as an high-level interface for converting entire domain names to and from the ACE encoded form. The TLD API consists of one set of functions to extract the TLD name from a domain string, one set of functions to locate the proper TLD table to use based on the TLD name, and core functions to validate a string against a TLD table, and some utility wrappers to perform all the steps in one call.

The library is used by, e.g., GNU SASL and Shishi to process user names and passwords. Libidn can be built into GNU Libc to enable a new system-wide getaddrinfo flag for IDN processing.

Libidn is developed for the GNU/Linux system, but runs on over 20 Unix platforms (including Solaris, IRIX, AIX, and Tru64) and Windows. The library is written in C and (parts of) the API is also accessible from C++, Emacs Lisp, Python and Java. A native Java and C# port is included.

Also included is a command line tool, several self tests, code examples, and more.

The internal layout of the library, and how your application interact with the various parts of the library, are shown in Figure 1.1.

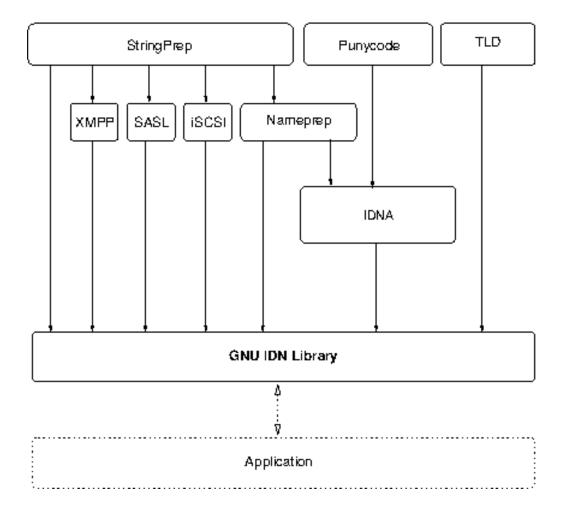


Figure 1.1: Components of Libidn

## 1.1 idna

idna —

## **Functions**

const char *	idna_strerror ()
int	idna_to_ascii_4i ()
int	idna_to_unicode_44i ()
int	idna_to_ascii_4z ()
int	idna_to_ascii_8z ()
int	idna_to_ascii_lz ()
int	idna_to_unicode_4z4z ()
int	idna_to_unicode_8z4z ()
int	idna_to_unicode_8z8z ()
int	idna_to_unicode_8zlz ()
int	idna_to_unicode_lzlz ()

## **Types and Values**

#define	IDNAPI
enum	Idna_rc
enum	Idna_flags
#define	IDNA ACE PREFIX

## **Description**

## **Functions**

## idna strerror ()

```
const char~*
idna_strerror (Idna_rc rc);
```

Convert a return code integer to a text string. This string can be used to output a diagnostic message to the user.

IDNA\_SUCCESS: Successful operation. This value is guaranteed to always be zero, the remaining ones are only guaranteed to hold non-zero values, for logical comparison purposes. IDNA\_STRINGPREP\_ERROR: Error during string preparation. IDNA\_PUNYCODE\_ERROR: Error during punycode operation. IDNA\_CONTAINS\_NON\_LDH: For IDNA\_USE\_STD3\_ASCII\_RU indicate that the string contains non-LDH ASCII characters. IDNA\_CONTAINS\_MINUS: For IDNA\_USE\_STD3\_ASCII\_RULES, indicate that the string contains a leading or trailing hyphen-minus (U+002D). IDNA\_INVALID\_LENGTH: The final output string is not within the (inclusive) range 1 to 63 characters. IDNA\_NO\_ACE\_PREFIX: The string does not contain the ACE prefix (for ToUnicode). IDNA\_ROUNDTRIP\_VERIFY\_ERROR: The ToASCII operation on output string does not equal the input. IDNA\_CONTAINS\_ACE\_PREFIX: The input contains the ACE prefix (for ToASCII). IDNA\_ICONV\_ERROR: Could not convert string in locale encoding. IDNA\_MALLOC\_ERROR: Could not allocate buffer (this is typically a fatal error). IDNA\_DLOPEN\_ERROR: Could not dlopen the libcidn DSO (only used internally in libc).

## **Parameters**

```
rc an Idna_rc return code.
```

#### Returns

Returns a pointer to a statically allocated string containing a description of the error with the return code rc.

## idna\_to\_ascii\_4i ()

The ToASCII operation takes a sequence of Unicode code points that make up one domain label and transforms it into a sequence of code points in the ASCII range (0..7F). If ToASCII succeeds, the original sequence and the resulting sequence are equivalent labels.

It is important to note that the ToASCII operation can fail. ToASCII fails if any step of it fails. If any step of the ToASCII operation fails on any label in a domain name, that domain name MUST NOT be used as an internationalized domain name. The method for deadling with this failure is application-specific.

The inputs to ToASCII are a sequence of code points, the AllowUnassigned flag, and the UseSTD3ASCIIRules flag. The output of ToASCII is either a sequence of ASCII code points or a failure condition.

ToASCII never alters a sequence of code points that are all in the ASCII range to begin with (although it could fail). Applying the ToASCII operation multiple times has exactly the same effect as applying it just once.

#### **Parameters**

in	input array with unicode	
111	code points.	
inlen	length of input array with	
illeli	unicode code points.	
	output zero terminated	
out	string that must have room	
out	for at least 63 characters	
	plus the terminating zero.	
	an Idna_flags value, e.g.,	
flags	IDNA_ALLOW_UNASSIGN	ED
nags	or	
	IDNA_USE_STD3_ASCII_R	ULES.

## **Returns**

Returns 0 on success, or an Idna\_rc error code.

## idna\_to\_unicode\_44i ()

The ToUnicode operation takes a sequence of Unicode code points that make up one domain label and returns a sequence of Unicode code points. If the input sequence is a label in ACE form, then the result is an equivalent internationalized label that is not in ACE form, otherwise the original sequence is returned unaltered.

ToUnicode never fails. If any step fails, then the original input sequence is returned immediately in that step.

The Punycode decoder can never output more code points than it inputs, but Nameprep can, and therefore ToUnicode can. Note that the number of octets needed to represent a sequence of code points depends on the particular character encoding used.

The inputs to ToUnicode are a sequence of code points, the AllowUnassigned flag, and the UseSTD3ASCIIRules flag. The output of ToUnicode is always a sequence of Unicode code points.

in	input array with unicode
III	code points.
inlen	length of input array with
inien	unicode code points.
out	output array with unicode
out	code points.
	on input, maximum size of
	output array with unicode
outlen	code points, on exit, actual
	size of output array with
	unicode code points.
	an Idna_flags value, e.g.,
flags	IDNA_ALLOW_UNASSIGNED
nags	or
	IDNA_USE_STD3_ASCII_RULES.

Returns Idna\_rc error condition, but it must only be used for debugging purposes. The output buffer is always guaranteed to contain the correct data according to the specification (sans malloc induced errors). NB! This means that you normally ignore the return code from this function, as checking it means breaking the standard.

## idna\_to\_ascii\_4z()

Convert UCS-4 domain name to ASCII string. The domain name may contain several labels, separated by dots. The output buffer must be deallocated by the caller.

#### **Parameters**

innut	zero terminated input	
input	Unicode string.	
output	pointer to newly allocated	
Output	output string.	
	an Idna_flags value, e.g.,	
flogs	IDNA_ALLOW_UNASSIGN	ED
flags	or	
	IDNA_USE_STD3_ASCII_R	ULES.

## **Returns**

Returns IDNA\_SUCCESS on success, or error code.

## idna\_to\_ascii\_8z ()

Convert UTF-8 domain name to ASCII string. The domain name may contain several labels, separated by dots. The output buffer must be deallocated by the caller.

input	zero terminated input UTF-8 string.	
output	pointer to newly allocated	
output	output string.	
	an Idna_flags value, e.g.,	
flags	IDNA_ALLOW_UNASSIGN	ED
flags	or	
	IDNA_USE_STD3_ASCII_R	ULES.

Returns IDNA\_SUCCESS on success, or error code.

## idna\_to\_ascii\_lz ()

Convert domain name in the locale's encoding to ASCII string. The domain name may contain several labels, separated by dots. The output buffer must be deallocated by the caller.

## **Parameters**

	zero terminated input string
input	encoded in the current
	locale's character set.
output	pointer to newly allocated
output	output string.
	an Idna_flags value, e.g.,
flags	IDNA_ALLOW_UNASSIGNED
nags	or
	IDNA_USE_STD3_ASCII_RULES.

## Returns

Returns IDNA\_SUCCESS on success, or error code.

## idna\_to\_unicode\_4z4z ()

Convert possibly ACE encoded domain name in UCS-4 format into a UCS-4 string. The domain name may contain several labels, separated by dots. The output buffer must be deallocated by the caller.

input	zero-terminated Unicode
mput	string.
output	pointer to newly allocated
output	output Unicode string.
	an Idna_flags value, e.g.,
flogs	IDNA_ALLOW_UNASSIGNED
flags	or
	IDNA_USE_STD3_ASCII_RULES.

Returns IDNA\_SUCCESS on success, or error code.

## idna\_to\_unicode\_8z4z ()

Convert possibly ACE encoded domain name in UTF-8 format into a UCS-4 string. The domain name may contain several labels, separated by dots. The output buffer must be deallocated by the caller.

## **Parameters**

input	zero-terminated UTF-8	
	string.	
output	pointer to newly allocated	
output	output Unicode string.	
	an Idna_flags value, e.g.,	
flags	IDNA_ALLOW_UNASSIGN	ED
	or	
	IDNA_USE_STD3_ASCII_R	ULES.

## Returns

Returns IDNA\_SUCCESS on success, or error code.

## idna\_to\_unicode\_8z8z ()

Convert possibly ACE encoded domain name in UTF-8 format into a UTF-8 string. The domain name may contain several labels, separated by dots. The output buffer must be deallocated by the caller.

#### **Parameters**

input	zero-terminated UTF-8	
	string.	
output	pointer to newly allocated	
output	output UTF-8 string.	
	an Idna_flags value, e.g.,	
flags	IDNA_ALLOW_UNASSIGN	ED
	or	
	IDNA_USE_STD3_ASCII_R	ULES.

#### **Returns**

Returns IDNA\_SUCCESS on success, or error code.

## idna\_to\_unicode\_8zlz ()

Convert possibly ACE encoded domain name in UTF-8 format into a string encoded in the current locale's character set. The domain name may contain several labels, separated by dots. The output buffer must be deallocated by the caller.

## **Parameters**

input	zero-terminated UTF-8	
	string.	
	pointer to newly allocated	
output	output string encoded in the	
σαιραί	current locale's character	
	set.	
	an Idna_flags value, e.g.,	
flags	IDNA_ALLOW_UNASSIGN	ED
	or	
	IDNA_USE_STD3_ASCII_R	ULES.

## Returns

Returns IDNA\_SUCCESS on success, or error code.

## idna\_to\_unicode\_lzlz()

Convert possibly ACE encoded domain name in the locale's character set into a string encoded in the current locale's character set. The domain name may contain several labels, separated by dots. The output buffer must be deallocated by the caller.

## **Parameters**

	zero-terminated string	
input	encoded in the current	
	locale's character set.	
	pointer to newly allocated	
output	output string encoded in the	
output	current locale's character	
	set.	
	an Idna_flags value, e.g.,	
flags	IDNA_ALLOW_UNASSIGN	ED
	or	
	IDNA_USE_STD3_ASCII_R	ULES.

## **Returns**

Returns IDNA\_SUCCESS on success, or error code.

## **Types and Values**

## **IDNAPI**

#define IDNAPI

## enum Idna\_rc

Enumerated return codes of idna\_to\_ascii\_4i(), idna\_to\_unicode\_44i() functions (and functions derived from those functions). The value 0 is guaranteed to always correspond to success.

## Members

IDNA\_SUCCESS

era tion. This value is guaranteed to ways be zero, the remaining ones are only guaranteed to hold nonzero values, for logical comparison purposes.

Successful

op-

	Error
	dur-
	ing
IDNA_STRINGPREP_ERROR	string
	prepa-
	ra-
	tion.
	Error
	dur-
	ing
	pun-
	y y t
IDNA_PUNYCODE_ERROR	code
	op-
	er-
	a- in-
	tion. For
	IDNA_USE_STD3_ASCII_RULES,
	in-
	di-
	cate
	that
	the
IDNA_CONTAINS_NON_LDH	string
IDINI_CONTRING_NON_EDIT	con-
	tains
	non-
	LDH
	ASCII
	char-
	ac-
	ters.

Sime as 2DV 2DV 2		
IDNA_CONTAINS_LIDH  IDNA_CONTAINS_LIDH  IDNA_CONTAINS_LIDH  IDNA_CONTAINS_LIDH  IDNA_USE_STD3_ASCII_RULES, in- discare that the string con- tiller that the string con- ti		Same
IDNA_CONTAINS_LIDH  IDNA_CONTAINS_LIDH  IDNA_CONTAINS_LIDH  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS		
IDNA_CONTAINS_LIDH  IDNA_CONTAINS_LIDH  IDNA_CONTAINS_LIDH  IDNA_CONTAINS_LIDH  IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- trail- ting or trail- ting or trail- ting or trail- ting or trail- ting hyphen- minus		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_MINUS		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS		A
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_MINUS		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_MINUS		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDH  IDH  IDH  IDH  IDH  IDH  IDH		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_MINUS		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_MINUS		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDH  IOPT  IDH  IDH  IUP  IUP  IUP  IUP  IUP  IUP  IUP  IU		
IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_LDH  IDNA_CONTAINS_MINUS		
IDNA_CONTAINS_LDH    Document		N
IDNA_CONTAINS_LDH    DH		
IDNA_CONTAINS_LDH    DH		
IDNA_CONTAINS_LDH    DH		N
IDNA_CONTAINS_LDH    Description   Descripti		
DH	IDNA CONTAINS I DU	
IDNA_CONTAINS_MINUS    IDNA_CONTAINS_MINUS	IDNA_CONTAINS_LDH	IV IV
for compating the compating of the compa		
for compatible for co		LDH
com- pat- i- bil- ity with typo in ear- lier Ver- sions.  For IDNA_USE_STD3_ASCH_RULES, in- di- cate that the string con- tinins a lead- ing or trail- ing hyphen- minus		,
ipati- i- bil- ity with typo in ear- lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- njinus		for
ipati- i- bil- ity with typo in ear- lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- njinus		com-
il- bil- ity with typo if ear- lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- minus		
bility with typo in ear- lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- minus		1   1
ity with typo in ear- lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tins a lead- ing or trail- ing hyphen- minus		
with typo in ear- lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tins a lead- ing or trail- ing hyphen- minus		
typo in ear- lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- trains a lead- ing or trail- ing hyphen- minus		
in ear- lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- minus		
ear- lier ver- sions.  For  IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- minus		typo
lier ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- minus		in
ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tiains a lead- ing or trail- ing hyphen- minus		ear-
ver- sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tiains a lead- ing or trail- ing hyphen- minus		lier
sions.  For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS  IDNA_CONTAINS_MINUS		
For IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- minus		
IDNA_USE_STD3_ASCII_RULES, in- di- cate that the string con- IDNA_CONTAINS_MINUS  tains a lead- ing or trail- ing hyphen- minus		
in- di- cate that the string con- tains a lead- ing or trail- ing hyphen- minus		
di- cate that the string con- tains a lead- ing or trail- ing hyphen- minus		
cate that the string con- IDNA_CONTAINS_MINUS  tains a lead- ing or trail- ing hyphen- minus		
that the string con- tains a lead- ing or trail- ing hyphen- minus		
the string contains_MINUS  tains a lead-ing or trail-ing hyphen-minus		
the string contains_MINUS  tains a lead-ing or trail-ing hyphen-minus		that
IDNA_CONTAINS_MINUS  tains a lead- ing or trail- ing hyphen- minus		the
IDNA_CONTAINS_MINUS  tains  a lead- ing or trail- ing hyphen- minus		string
IDNA_CONTAINS_MINUS  tains a lead- ing or trail- ing hyphen- minus		con-
a lead- ing or trail- ing hyphen- minus	IDNA CONTAINS MINUS	tains
lead- ing or trail- ing hyphen- minus	121/11_001111111100	
ing or trail- ing hyphen- minus		
or trail- ing hyphen- minus		15au-   .
or trail- ing hyphen- minus		ıng
ing hyphen- minus		or
ing hyphen- minus		trail-
minus		ing
minus		hyphen-
ninus		minus
41.000		(U+002D).
(Ψ+002D).		(ψ+002D).

	The
	fi <sub>-</sub>
	nal
	out-
	put
	string
	is
	not
	within
	the
IDNA_INVALID_LENGTH	(in-
	clu-
	sive)
	range
	1
	to
	63
	char-
	a¢-
	ters.
	The
	string
	does
	not
	con-
	tạin
IDNA_NO_ACE_PREFIX	the
	ACE
	pre-
	fix
	(for
	ToUni-
	code).
	The
	ToASCII
	op-
	er-
	a-
	tion
	on
	out-
IDNA_ROUNDTRIP_VERIFY_ERROR	
	put
	string
	does
	not
	equal
	the
	uic
	in-
	put.
	1 * 1

IDNA_CONTAINS_ACE_PREFIX	The input contains the ACE prefix (for ToASCII).
IDNA_ICONV_ERROR	Could not convert string in lo-cale en-cod-ing.
IDNA_MALLOC_ERROR	Could not al- lo- cate buffer (this is typ- i- cally a fa- tal er- ror).
IDNA_DLOPEN_ERROR	Could not dlopen the lib-cidn DSO (only used internally in libc).

## enum Idna\_flags

Flags to pass to idna\_to\_ascii\_4i(), idna\_to\_unicode\_44i() etc.

## **Members**

IDNA_ALLOW_UNASSIGNED	Don't re- ject strings con- tain- ing unas- signed Uni- code code points.
IDNA_USE_STD3_ASCII_RULES	Validate strings ac- cord- ing to STD3 rules (i.e., nor- mal host name rules).

## IDNA\_ACE\_PREFIX

```
# define IDNA_ACE_PREFIX "xn--"
```

The IANA allocated prefix to use for IDNA. "xn--"

## 1.2 stringprep

stringprep —

## **Types and Values**

#define	IDNAPI
#define	STRINGPREP_VERSION
enum	Stringprep_rc
enum	Stringprep_profile_flags
enum	Stringprep_profile_steps
#define	STRINGPREP_MAX_MAP_CHARS

## **Description**

## **Functions**

## **Types and Values**

## **IDNAPI**

#define

IDNAPI

## STRINGPREP\_VERSION

```
# define STRINGPREP_VERSION "1.30"
```

String defined via CPP denoting the header file version number. Used together with <a href="stringprep\_check\_version">stringprep\_check\_version</a>() to verify header file and run-time library consistency.

## enum Stringprep\_rc

Enumerated return codes of stringprep(), stringprep\_profile() functions (and macros using those functions). The value 0 is guaranteed to always correspond to success.

## **Members**

	Successful
	op-
	er-
	a
	tion.
	This
	value
	is
	guar-
	an-
	teed
	to
	al-
	ways
	be
	zero,
	the
	re-
	main-
	ing
STRINGPREP_OK	ones
STRINGTREE _OR	are
	only
	guar-
	an-
	teed
	to
	hold
	non-
	zero
	val-
	ues,
	for
	log-
	i-
	cal
	com-
	par-
	par- i-
	son
	pur-
	poses.

STRINGPREP_CONTAINS_UNASSIGNED  STRINGPREP_CONTAINS_UNASSIGNED  STRINGPREP_CONTAINS_UNASSIGNED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED		
contain unassigned Unicode code code code points, which is for- bid- den by tite pro- file.  String contain tin code points String con- titin code points STRINGPREP_CONTAINS_PROHIBITED  STRI		String
stringprep_contains_unassigned  Uni- code code points. which is for- bid- den by the pro- file.  Stringprep_contains_prohibited  STRINGPREP_CONTAINS_PROHIBITE		
uinas- signed Uni- code code code points, which is for- bid- den by the pro- file.  String con- tain code points by the pro- hib- iged by the pro- file.  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  Fire- ting con- tain code pionts with con- filec- ting bidit- rec- tion cat- e-		
signed Uni- code code code points, which is for- bid- den by ttle pro- file.  String con- tain code points stain code points is string con- tain code points ite by ttle pro- file.  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  Gen- tinn code points with con- tain code points with code po		
STRINGPREP_CONTAINS_UNASSIGNED  Points, which is for-bid-den by the pro-file.  STring contain code points stringPrep_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  Con-dilution code points with code code code code code code code code		
STRINGPREP_CONTAINS_UNASSIGNED  Points, which is for-bid-den by the pro-file.  STring contain code points stringPrep_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  Con-dilution code points with code code code code code code code code		signed
STRINGPREP_CONTAINS_UNASSIGNED  STRINGPREP_CONTAINS_UNASSIGNED  String contain code points String contain code points STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDI_BOTH_L_AND_RAL		Uni-
STRINGPREP_CONTAINS_UNASSIGNED  points, which is for- bid- den by the pro- file.  String con- tain code points  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL  STRINGPREP_BIDL_BOTH_L_AND_RAL		
STRINGPREP_CONTAINS_UNASSIGNED    points, which is for-bid-den by the profile.    String contains code points prohib-lited by the profile.    STRINGPREP_CONTAINS_PROHIBITED   Prohib-lited by the profile.    String contains prohib-lited by the profile.    String contains prohib-lited by the profile.    String contain code points with constain code points with code		
STRINGPREP_CONTAINS_UNASSIGNED  which is for- bid- den by the pro- file.  String con- tain code points pro- hib- ited by the pro- file.  String con- tain code pints ited by the pro- file.  String con- tain code pints ited by the pro- file.  String con- tain code points string con- tain code points with con- tain code		
STRINGPREP_BIDI_BOTH_L_AND_RAL    String contains code points   String contain code	STRINGPREP CONTAINS LINASSIGNED	points,
for- bjd- den by the pro- file.  String con- tain code points pro- hib- ited by ttie pro- file.  String con- tain code points string tied by ttie pro- file.  String con- tain code pro- file.  String con- tain code points string con- tain code points with code p	STRINGTREE _CONTINUO_ON/ISSIGNED	which
for- bjd- den by the pro- file.  String con- tain code points pro- hib- ited by ttie pro- file.  String con- tain code points string tied by ttie pro- file.  String con- tain code pro- file.  String con- tain code points string con- tain code points with code p		is
bid- den bby tthe pro- file.  String con- tain code points pro- hib- ited by tthe pro- file.  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  File.  STRINGPREP_BIDI_BOTH_L_AND_RAL  File.  STRINGPREP_BIDI_BOTH_L_AND_RAL  File.  STRINGPREP_BIDI_BOTH_L_AND_RAL  File.  STRINGPREP_BIDI_BOTH_L_AND_RAL  File.  STRINGPREP_BIDI_BOTH_L_AND_RAL		
den by the pro- file.  String con- tain code points pro- hib- ited by the pro- file.  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  Gen tain code points with con- tain code points wi		
by the pro- file.  String con- tain code points pro- file-  STRINGPREP_CONTAINS_PROHIBITED  Pro- file- ited by the pro- file.  SIring con- tain code points with con- tain code		
the profile.  String contain code points  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  Prohib- ited by the profile.  String contain code points with con- filit- ing bidi- rec- tion cat- e-		
string contain code points  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL		by
string contain code points  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL		the
file.  String con- tain code points pro- hib- ited by tthe pro- file.  String con- tain code points String con- tain code pro- file.  String con- tain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  filet- ing bidi- rec- tion cat- e-		
String con- tain code points pro- hib- ited by the pro- file.  String con- tain code points String con- file.  String con- tain code points with con- filet- ing bidi- rec- tion cat- e-		
STRINGPREP_CONTAINS_PROHIBITED  STRINGPREP_CONTAINS_PROHIBITED  Prohib- ited by the pro- file.  String con- tain code points with con- flict- ing bidi- rec- tion cat- e-		
STRINGPREP_CONTAINS_PROHIBITED  pro- hib- ited by the pro- file.  String con- tain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-		
STRINGPREP_CONTAINS_PROHIBITED  pro- hib- ited by the pro- file.  String con- tain code points with con- tain code points with con- tain code points with cen- tion cat- e- tion cat- e-		con-
STRINGPREP_CONTAINS_PROHIBITED  pro- hib- ited by the pro- file.  String con- tain code points with con- tain code points with con- tain code points with cen- tion cat- e- tion cat- e-		tain
STRINGPREP_CONTAINS_PROHIBITED  pro- hib- ited by tthe pro- file.  String con- tain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  pro- file.  String cot- ting points with con- filct- ing bidi- rec- tion cat- e-		
STRINGPREP_CONTAINS_PROHIBITED  prohib- ited by the pro- file.  String con- tain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-		
hib- ited by the pro- file.  String con- tain code points with con- flict- ing bidi- rec- tion cat- e-		
ited by the pro- file.  String con- tain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-	STRINGPREP CONTAINS PROHIBITED	
by the profile.  String contain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL		hib-
by the profile.  String contain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL  STRINGPREP_BIDI_BOTH_L_AND_RAL		ited
the profile.  String contain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-		
profile.  String contain code points with con- flict- ing bidi- rec- tion cat- e-		
file.  String  contain  code  points  with  conflict-  ing  bidi- rec- tion  cat- e-		
String con- tain code points with con- flict- ing bidi- rec- tion cat- e-		
contain code points with con- flict- ing bidi- rec- tion cat- e-		
contain code points with con- flict- ing bidi- rec- tion cat- e-		String
tain code points with con- STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-		
code points with con- flict- ing bidi- rec- tion cat- e-		
points with con- flict- ing bidi- rec- tion cat- e-		
STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-		
STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-		
STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-		with
STRINGPREP_BIDI_BOTH_L_AND_RAL  flict- ing bidi- rec- tion cat- e-		con-
ing bidi- rec- tion cat- e-	STRINGPREP RIDI BOTH L AND RAL	flict-
bidi- rec- tion cat- e-		ing
rec- tion cat- e-		
tion cat- e-		
cat-		rec-
cat-		tion
e-		
gory.		
		gory.

	Leading
	and
	trail-
	ing
	char-
	ac-
	ter
	in
STRINGPREP_BIDI_LEADTRAIL_NOT_RAL	string
STRINGT REF_DIDT_EEFED TRAILE_NOT_RAIL	not
	of
	proper
	bidi-
	rec-
	tional
	cat-
	e <del>+</del>
	gory.
	Contains
	pro-
	hib-
	ited
	code
	points
STRINGPREP_BIDI_CONTAINS_PROHIBITED	de-
	tected
	by
	bidi-
	rec-
	tional
	code.
	Buffer
	handed
	to
	func-
	tion
	was
	too
	small.
	Silian.
	This
	usu-
	ally
	in-
STRINGPREP_TOO_SMALL_BUFFER	di-
	cate
	prob-
	lem
	in
	the
	call-
	ing
	ap-
	ap- pli-
	Pµ-
	ca-
	1
	tion.

	The
	string-
	prep
	pro-
	file
	was
	in-
	con-
	sis-
	tent.
	This
	usu-
CTRINCIPED DROEILE EDROR	ally
STRINGPREP_PROFILE_ERROR	in-
	di-
	cate
	an
	in-
	ter-
	nal
	er-
	ror
	in
	the
	li-
	brary.
	The
	sup-
	plied
	flag
	con-
	flicted
	with
	pro-
	pio-
	file.
	This
	usu-
	ally
	in-
STRINGPREP_FLAG_ERROR	
	di-
	cate
	a
	prob-
	lem
	in
	the
	call-
	ing
	ap-
	pli-
	ca-
	tion.

	The
	sup-
	plied
	pro- file
OTEN MADDED ANAMAMAN DECEME	name
STRINGPREP_UNKNOWN_PROFILE	was
	not
	known
	to
	the
	li-
	brary.
	The
	Uni-
	code
	NFKC
	op-
	er-
	a
	tion
	failed.
	This
	uşu-
CTDINCEDED NEW CEATIED	ally
STRINGPREP_NFKC_FAILED	in-
	di-
	cate
	an
	in-
	ter-
	nal
	er-
	ror
	in
	the
	li-
	li- brary.
	The
	mal-
	loc()
	was
	out
	of
	mem-
	ory. This
STRINGPREP_MALLOC_ERROR	This
	is
	usu-
	ally
	a
	fa-
	tal
	er-
	ror.
	1 1 1

enum Stringprep profile flags	enum	Stringprep	profile	flags
-------------------------------	------	------------	---------	-------

Stringprep profile flags.

Members

Disable the NFKC normalization, as well as se-lect-ing the non-NFKC case folding tables. Usually the profile STRINGPREP\_NO\_NFKC specifies BIDI and NFKC settings, and applications should not override it unless in speclal situ a tions.

	Disable the BIDI step. Usu- ally the pro- file spec- i-
STRINGPREP_NO_BIDI	set- tings, and ap- pli- ca- tions should not over- ride it un-
	less in spe- cial sit- u- a- tions.

	Laker
	Make
	the
	li-
	brary
	re-
	turn
	with
	an
	er-
	ror
	if
	string
STRINGPREP_NO_UNASSIGNED	con-
	tains
	unas-
	signed
	char-
	ac-
	ters
	ac-
	cord-
	ing
	to
	pro-
	file.
	1 1

## enum Stringprep\_profile\_steps

Various steps in the stringprep algorithm. You really want to study the source code to understand this one. Only useful if you want to add another profile.

## Members

STRINGPREP_NFKC	The NFKC
	step.
	The
STRINGPREP_BIDI	BIDI
	step.
	The
STRINGPREP_MAP_TABLE	MAP
	step.
	The
STRINGPREP_UNASSIGNED_TABLE	Unas-
STRINGTREE _OTWISSIGNED_TRIBEE	signed
	step.
	The
	Pro-
STRINGPREP_PROHIBIT_TABLE	hib-
	ited
	step.
	The
STRINGPREP_BIDI_PROHIBIT_TABLE	BIDI-
oranioral publication in the control of the control and the control of the contro	Prohibited
	step.

STRINGPREP_BIDI_RAL_TABLE	The BIDI- RAL step.
STRINGPREP_BIDI_L_TABLE	The BIDI- L step.

## STRINGPREP\_MAX\_MAP\_CHARS

```
# define STRINGPREP_MAX_MAP_CHARS 4
```

Maximum number of code points that can replace a single code point, during stringprep mapping.

## 1.3 punycode

punycode —

## **Functions**

const char *	punycode_strerror ()
int	punycode_encode ()
int	punycode_decode ()

## **Types and Values**

#define	IDNAPI
enum	Punycode_status
typedef	punycode uint

## **Description**

## **Functions**

## punycode\_strerror ()

```
const char~*
punycode_strerror (Punycode_status rc);
```

Convert a return code integer to a text string. This string can be used to output a diagnostic message to the user.

PUNYCODE\_SUCCESS: Successful operation. This value is guaranteed to always be zero, the remaining ones are only guaranteed to hold non-zero values, for logical comparison purposes. PUNYCODE\_BAD\_INPUT: Input is invalid. PUNYCODE\_BIG\_OUTPUT: Output would exceed the space provided. PUNYCODE\_OVERFLOW: Input needs wider integers to process.

## **Parameters**

rc an Punycode\_status return code.

Returns a pointer to a statically allocated string containing a description of the error with the return code rc.

## punycode\_encode ()

Converts a sequence of code points (presumed to be Unicode code points) to Punycode.

	The number of code points
input_length	in the input array and the
	number of flags in the
	case_flags array.
	An array of code points.
	They are presumed to be
	Unicode code points, but
	that is not strictly
	REQUIRED. The array
	contains code points, not
	code units. UTF-16 uses
	code units D800 through
inaut	DFFF to refer to code
input	points 1000010FFFF. The
	code points D800DFFF do
	not occur in any valid
	Unicode string. The code
	points that can occur in
	Unicode strings (0D7FF
	and E00010FFFF) are also
	called Unicode scalar
	values.

case_flags	A NULL pointer or an array of boolean values parallel to the <i>input</i> array. Nonzero (true, flagged) suggests that the corresponding Unicode character be forced to uppercase after being decoded (if possible), and zero (false, unflagged) suggests that it be forced to lowercase (if possible).  ASCII code points (07F) are encoded literally, except that ASCII letters are forced to uppercase or	
	lowercase according to the corresponding case flags. If case_flags is a NULL pointer then ASCII letters are left as they are, and other code points are treated as unflagged.	
output_length	The caller passes in the maximum number of ASCII code points that it can receive. On successful return it will contain the number of ASCII code points actually output.	
output	An array of ASCII code points. It is *not* null-terminated; it will contain zeros if and only if the input contains zeros. (Of course the caller can leave room for a terminator and add one if needed.)	

The return value can be any of the Punycode\_status values defined above except PUNYCODE\_BAD\_INPUT. If not PUNYCODE\_SUCCESS, then <code>output\_size</code> and <code>output</code> might contain garbage.

## punycode\_decode ()

Converts Punycode to a sequence of code points (presumed to be Unicode code points).

input_length	The number of ASCII code	
mput_iengui	points in the input array.	
input	An array of ASCII code	
mput	points (07F).	
	The caller passes in the	
	maximum number of code	
	points that it can receive	
	into the output array	
	(which is also the	
	maximum number of flags	
	that it can receive into the	
	case_flags array, if	
	case_flags is not a	
	NULL pointer). On	
	successful return it will	
	contain the number of code	
	points actually output	
	(which is also the number	
output_length	of flags actually output, if	
	case_flags is not a null	
	pointer). The decoder will	
	never need to output more	
	code points than the number	
	of ASCII code points in the	
	input, because of the way	
	the encoding is defined.	
	The number of code points	
	output cannot exceed the	
	maximum possible value of	
	a punycode_uint, even if the	
	supplied output_length	
	is greater than that.	
	An array of code points like	
output	the input argument of	
1	punycode_encode() (see	
	above).	
	A NULL pointer (if the	
	flags are not needed by the	
case_flags	caller) or an array of	
	boolean values parallel to	
	the output array. Nonzero	
	(true, flagged) suggests that	
	the corresponding Unicode	
	character be forced to	
	uppercase by the caller (if	
	possible), and zero (false,	
	unflagged) suggests that it	
	be forced to lowercase (if	
	possible). ASCII code	
	points (07F) are output	
	already in the proper case,	
	but their flags will be set	
	appropriately so that	
	applying the flags would be	
	harmless.	

The return value can be any of the Punycode\_status values defined above. If not PUNYCODE\_SUCCESS, then  $output\_length$ , output, and  $case\_flags$  might contain garbage.

## **Types and Values**

## **IDNAPI**

#define

IDNAPI

## enum Punycode\_status

Enumerated return codes of punycode\_encode() and punycode\_decode(). The value 0 is guaranteed to always correspond to success.

## **Members**

	Successful
	op-
	er-
	a
	tion.
	This
	value
	value
	is
	guar-
	an-
	teed
	to
	al-
	ways
	be
	zero,
	the
	re-
	main-
	ing
	ones
PUNYCODE_SUCCESS	are
	only
	guar-
	an-
	teed
	to
	hold
	non-
	zero
	val-
	ues,
	for
	log-
	i-
	cal
	com-
	par-   i-
	søn
	pur-
	poses.
	Input
DIDIVOODE DAD DIDIVE	is
PUNYCODE_BAD_INPUT	in-
	valid.
	Output
	would
	ex-
PUNYCODE_BIG_OUTPUT	ceed
TOM TOODE_DIO_OUTFUL	the
	space
	pro-
	vided.
	viaca.

PUNYCODE\_OVERFLOW

PUNYCODE\_OVERFLOW

tegers
to
process.

## punycode\_uint

```
typedef uint32_t punycode_uint;
```

Unicode code point data type, this is always a 32 bit unsigned integer.

## 1.4 pr29

pr29 —

## **Functions**

const char *	pr29_strerror ()
int	pr29_4 ()
int	pr29_4z ()
int	pr29_8z ()

## **Types and Values**

#define	IDNAPI
enum	Pr29_rc

## **Description**

## **Functions**

## pr29\_strerror()

```
const char~*
pr29_strerror (Pr29_rc rc);
```

Convert a return code integer to a text string. This string can be used to output a diagnostic message to the user.

PR29\_SUCCESS: Successful operation. This value is guaranteed to always be zero, the remaining ones are only guaranteed to hold non-zero values, for logical comparison purposes. PR29\_PROBLEM: A problem sequence was encountered. PR29\_STRINGPREP\_ERROR: The character set conversion failed (only for pr29\_8z()).

## **Parameters**

rc an Pr29\_rc return code.

Returns a pointer to a statically allocated string containing a description of the error with the return code rc.

## pr29\_4 ()

Check the input to see if it may be normalized into different strings by different NFKC implementations, due to an anomaly in the NFKC specifications.

## **Parameters**

in	input array with unicode code points.	
len	length of input array with	
icii	unicode code points.	

#### Returns

Returns the Pr29\_rc value PR29\_SUCCESS on success, and PR29\_PROBLEM if the input sequence is a "problem sequence" (i.e., may be normalized into different strings by different implementations).

## pr29\_4z()

```
int
pr29_4z (const uint32_t *in);
```

Check the input to see if it may be normalized into different strings by different NFKC implementations, due to an anomaly in the NFKC specifications.

## **Parameters**

in	zero terminated array of
	Unicode code points.

## Returns

Returns the Pr29\_rc value PR29\_SUCCESS on success, and PR29\_PROBLEM if the input sequence is a "problem sequence" (i.e., may be normalized into different strings by different implementations).

## pr29\_8z ()

```
int
pr29_8z (const char *in);
```

Check the input to see if it may be normalized into different strings by different NFKC implementations, due to an anomaly in the NFKC specifications.

in	zero terminated input
ın	UTF-8 string.

Returns the Pr29\_rc value PR29\_SUCCESS on success, and PR29\_PROBLEM if the input sequence is a "problem sequence" (i.e., may be normalized into different strings by different implementations), or PR29\_STRINGPREP\_ERROR if there was a problem converting the string from UTF-8 to UCS-4.

## **Types and Values**

## **IDNAPI**

## enum Pr29\_rc

Enumerated return codes for pr29\_4(), pr29\_4z(), pr29\_8z(). The value 0 is guaranteed to always correspond to success.

## **Members**

	d c.
	Successful
	р-
	r-
a	†
ti	ion.
T	his
V	alue
is	
g	uar-
	n-
te	eed
to	o
a	.1-
w w	vays
b	p <b>e</b>
	ero,
	he
	e-
n	nain-
ir	ng
	nes
	re
	phly
	uar-
a	n-
te	eed
to	
h h	old
	on-
	ero
	al-
	es,
f	or
	og-
i-	<u> </u>
	al
	om-
<i>p</i>   i-	par-
	on
l p	our-
	oses.
A	1 vah
$\mid \stackrel{p}{\underset{1}{\cdot}}$	rob-
	em 
	e-
	uence
	vas
	n-
	oun-
te	ered.

PR29\_STRINGPREP\_ERROR

The character set conversion failed (only for pr29\_8z()).

## 1.5 tld

tld —

## **Types and Values**

#define | IDNAPI

**Description** 

**Functions** 

**Types and Values** 

IDNAPI

#define IDNAPI

## 1.6 idn-free

idn-free —

## **Types and Values**

#define IDNAPI

**Description** 

**Functions** 

**Types and Values** 

**IDNAPI** 

#define IDNAPI

## **Chapter 2**

## Index

```
IDNA_ACE_PREFIX, 14
Idna_flags, 13
Idna_rc, 9
idna_strerror, 3
idna_to_ascii_4i, 3
idna_to_ascii_4z, 5
idna_to_ascii_8z, 5
idna_to_ascii_lz, 6
idna_to_unicode_44i, 4
idna_to_unicode_4z4z, 6
idna_to_unicode_8z4z, 7
idna_to_unicode_8z8z, 7
idna_to_unicode_8zlz, 8
idna_to_unicode_lzlz, 8
IDNAPI, 9, 15, 29, 34, 36
P
pr29_4, 32
pr29_4z, 32
pr29_8z, 32
Pr29_rc, 34
pr29_strerror, 31
punycode_decode, 27
punycode_encode, 26
Punycode_status, 29
punycode_strerror, 25
punycode_uint, 31
STRINGPREP_MAX_MAP_CHARS, 25
Stringprep_profile_flags, 21
Stringprep_profile_steps, 24
Stringprep_rc, 15
STRINGPREP_VERSION, 15
```