| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/IDN.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/net/HttpURLConnection.html)   [**NEXT CLASS**](http://docs.google.com/java/net/Inet4Address.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/IDN.html)    [**NO FRAMES**](http://docs.google.com/IDN.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#2s8eyo1) |

## **java.net**

Class IDN

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.net.IDN**

public final class **IDN**extends [Object](http://docs.google.com/java/lang/Object.html)

Provides methods to convert internationalized domain names (IDNs) between a normal Unicode representation and an ASCII Compatible Encoding (ACE) representation. Internationalized domain names can use characters from the entire range of Unicode, while traditional domain names are restricted to ASCII characters. ACE is an encoding of Unicode strings that uses only ASCII characters and can be used with software (such as the Domain Name System) that only understands traditional domain names.

Internationalized domain names are defined in [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt). RFC 3490 defines two operations: ToASCII and ToUnicode. These 2 operations employ [Nameprep](http://www.ietf.org/rfc/rfc3491.txt) algorithm, which is a profile of [Stringprep](http://www.ietf.org/rfc/rfc3454.txt), and [Punycode](http://www.ietf.org/rfc/rfc3492.txt) algorithm to convert domain name string back and forth.

The behavior of aforementioned conversion process can be adjusted by various flags:

* If the ALLOW\_UNASSIGNED flag is used, the domain name string to be converted can contain code points that are unassigned in Unicode 3.2, which is the Unicode version on which IDN conversion is based. If the flag is not used, the presence of such unassigned code points is treated as an error.
* If the USE\_STD3\_ASCII\_RULES flag is used, ASCII strings are checked against [RFC 1122](http://www.ietf.org/rfc/rfc1122.txt) and [RFC 1123](http://www.ietf.org/rfc/rfc1123.txt). It is an error if they don't meet the requirements.

These flags can be logically OR'ed together.

The security consideration is important with respect to internationalization domain name support. For example, English domain names may be *homographed* - maliciously misspelled by substitution of non-Latin letters. [Unicode Technical Report #36](http://www.unicode.org/reports/tr36/) discusses security issues of IDN support as well as possible solutions. Applications are responsible for taking adequate security measures when using international domain names.

**Since:** 1.6

| **Field Summary** | |
| --- | --- |
| static int | [**ALLOW\_UNASSIGNED**](http://docs.google.com/java/net/IDN.html#ALLOW_UNASSIGNED)            Flag to allow processing of unassigned code points |
| static int | [**USE\_STD3\_ASCII\_RULES**](http://docs.google.com/java/net/IDN.html#USE_STD3_ASCII_RULES)            Flag to turn on the check against STD-3 ASCII rules |

| **Method Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**toASCII**](http://docs.google.com/java/net/IDN.html#toASCII(java.lang.String))([String](http://docs.google.com/java/lang/String.html) input)            Translates a string from Unicode to ASCII Compatible Encoding (ACE), as defined by the ToASCII operation of [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt). |
| static [String](http://docs.google.com/java/lang/String.html) | [**toASCII**](http://docs.google.com/java/net/IDN.html#toASCII(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) input, int flag)            Translates a string from Unicode to ASCII Compatible Encoding (ACE), as defined by the ToASCII operation of [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt). |
| static [String](http://docs.google.com/java/lang/String.html) | [**toUnicode**](http://docs.google.com/java/net/IDN.html#toUnicode(java.lang.String))([String](http://docs.google.com/java/lang/String.html) input)            Translates a string from ASCII Compatible Encoding (ACE) to Unicode, as defined by the ToUnicode operation of [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt). |
| static [String](http://docs.google.com/java/lang/String.html) | [**toUnicode**](http://docs.google.com/java/net/IDN.html#toUnicode(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) input, int flag)            Translates a string from ASCII Compatible Encoding (ACE) to Unicode, as defined by the ToUnicode operation of [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt). |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### ALLOW\_UNASSIGNED

public static final int **ALLOW\_UNASSIGNED**

Flag to allow processing of unassigned code points

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.net.IDN.ALLOW_UNASSIGNED)

### USE\_STD3\_ASCII\_RULES

public static final int **USE\_STD3\_ASCII\_RULES**

Flag to turn on the check against STD-3 ASCII rules

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.net.IDN.USE_STD3_ASCII_RULES)

| **Method Detail** |
| --- |

### toASCII

public static [String](http://docs.google.com/java/lang/String.html) **toASCII**([String](http://docs.google.com/java/lang/String.html) input,  
 int flag)

Translates a string from Unicode to ASCII Compatible Encoding (ACE), as defined by the ToASCII operation of [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt).

ToASCII operation can fail. ToASCII fails if any step of it fails. If ToASCII operation fails, an IllegalArgumentException will be thrown. In this case, the input string should not be used in an internationalized domain name.

A label is an individual part of a domain name. The original ToASCII operation, as defined in RFC 3490, only operates on a single label. This method can handle both label and entire domain name, by assuming that labels in a domain name are always separated by dots. The following characters are recognized as dots: \u002E (full stop), \u3002 (ideographic full stop), \uFF0E (fullwidth full stop), and \uFF61 (halfwidth ideographic full stop). if dots are used as label separators, this method also changes all of them to \u002E (full stop) in output translated string.

**Parameters:**input - the string to be processedflag - process flag; can be 0 or any logical OR of possible flags **Returns:**the translated String **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the input string doesn't conform to RFC 3490 specification

### toASCII

public static [String](http://docs.google.com/java/lang/String.html) **toASCII**([String](http://docs.google.com/java/lang/String.html) input)

Translates a string from Unicode to ASCII Compatible Encoding (ACE), as defined by the ToASCII operation of [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt).

This convenience method works as if by invoking the two-argument counterpart as follows:

[toASCII](http://docs.google.com/java/net/IDN.html#toASCII(java.lang.String,%20int))(input, 0);

**Parameters:**input - the string to be processed **Returns:**the translated String **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the input string doesn't conform to RFC 3490 specification

### toUnicode

public static [String](http://docs.google.com/java/lang/String.html) **toUnicode**([String](http://docs.google.com/java/lang/String.html) input,  
 int flag)

Translates a string from ASCII Compatible Encoding (ACE) to Unicode, as defined by the ToUnicode operation of [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt).

ToUnicode never fails. In case of any error, the input string is returned unmodified.

A label is an individual part of a domain name. The original ToUnicode operation, as defined in RFC 3490, only operates on a single label. This method can handle both label and entire domain name, by assuming that labels in a domain name are always separated by dots. The following characters are recognized as dots: \u002E (full stop), \u3002 (ideographic full stop), \uFF0E (fullwidth full stop), and \uFF61 (halfwidth ideographic full stop).

**Parameters:**input - the string to be processedflag - process flag; can be 0 or any logical OR of possible flags **Returns:**the translated String

### toUnicode

public static [String](http://docs.google.com/java/lang/String.html) **toUnicode**([String](http://docs.google.com/java/lang/String.html) input)

Translates a string from ASCII Compatible Encoding (ACE) to Unicode, as defined by the ToUnicode operation of [RFC 3490](http://www.ietf.org/rfc/rfc3490.txt).

This convenience method works as if by invoking the two-argument counterpart as follows:

[toUnicode](http://docs.google.com/java/net/IDN.html#toUnicode(java.lang.String,%20int))(input, 0);

**Parameters:**input - the string to be processed **Returns:**the translated String

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/IDN.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/net/HttpURLConnection.html)   [**NEXT CLASS**](http://docs.google.com/java/net/Inet4Address.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/IDN.html)    [**NO FRAMES**](http://docs.google.com/IDN.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#2s8eyo1) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).