D.5 IDNA 2003 Compatibility

This appendix describes two sets of code point which are related to the compatibility between IDNA 2003 and IDNA 2008. Section D.5.1 focuses on the Latin Letter Sharp S (ß) U+00DF while Section D.5.2 focuses on the Latin Small Letter Dotless I (ı) U+0131. For each section, the difference in browser behavior and the user experience are analyzed.

D.5.1 Latin Small Letter Sharp S (ß) 00DF

IDNA 2003 Versus IDNA 2008

One of the differences between IDNA 2008 and IDNA 2003 is the treatment of four characters, one of which is relevant to the Latin Script LGR: the Latin Small Letter Sharp S or 00DF. Despite the fact IDNA 2008 superseded IDNA 2003, some applications continued to apply the character mapping from IDNA2003, resulting in DNS lookup queries that look like the following:

Table D.1. DNS resolution comparison for Sharp S (00DF)

Char	Example	IDNA 2003 Result	IDNA 2008 Result
ß 00DF	href="**http://faß.de"**	**http://faß.de** → **http://fass.de**	**http://faß.de** → **http://xn--fa-hia.de**

Source: https://unicode.org/reports/tr46/#Transition_Considerations

The difference in application behavior relative to DNS labels containing the code point 00DF causes two types of problems:

1. **Failure of service**. The user intends to navigate to “example.faß” but the application sends the user to “example.fass” which doesn’t exist, because the domain name is not registered or is blocked or withheld.
2. **Misconnection**. The user intends to navigate to “example.faß” but the browser returns “example.fass” which is controlled by a different registrant.

Internet Browser Support

As of the writing of this proposal, certain Internet browsers process 00DF using the IDNA 2003 mapping mechanism, instead of doing the IDNA 2008 conversion. A test with the four major Internet browsers shows that Google Chrome and Microsoft Edge have not fully implemented IDNA 2008; they still are in what is called “transitional mode”. For more information about

IDNA 2008 transitional mode, see Unicode Technical Standard #46 at https://unicode.org/reports/tr46/.

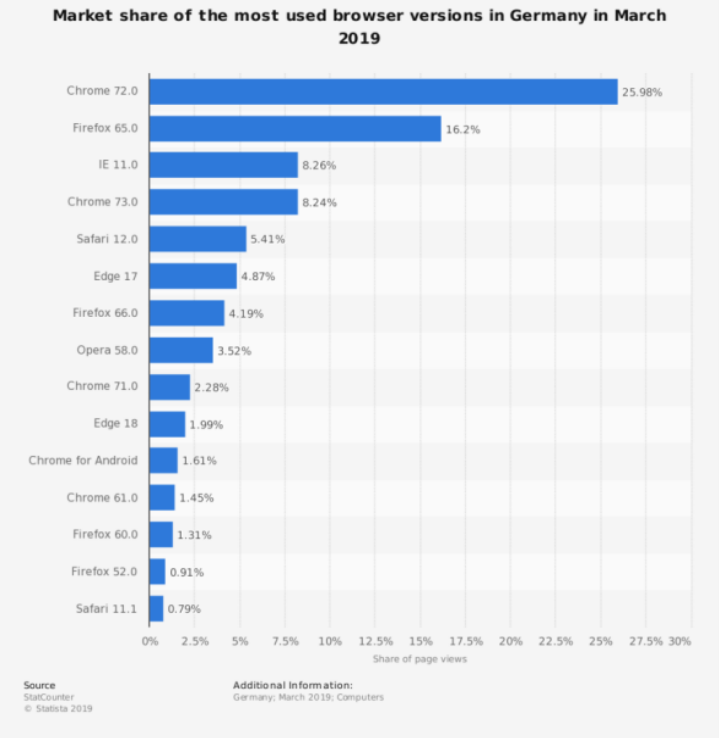
Table D.2. Resolution of **http://faß.de** by Different Internet Browsers

Internet Browser	**http://faß.de resolves to**
Microsoft Edge/Explorer	**http://fass.de**
Apple Safari	**http://xn--fa-hia.de**
Firefox	**http://xn--fa-hia.de**
Google Chrome	**http://fass.de**

The trend of browser implementation seems to be towards full IDNA 2008 compliance (given that Apple Safari and Firefox did migrate from IDNA 2003 to IDNA 2008). However, it is not clear how soon or late Google Chrome or Microsoft Edge will fully transition to IDNA 2008. See for example, https://bugs.chromium.org/p/chromium/issues/detail?id=941691

As of March 2019, Chrome has the largest browser market share in Germany, which suggests an important part of the end-user population is exposed to the problem with DNS lookups when utilizing the non-IDNA 2008-conforming browsers when the label contains code point 00DF.

Diagram D.2: Market Share of the Most Used Browser Versions in Germany in March 2019



Registry Implementation at the Second Level
Latin GP sought the input of TLD registries serving the German-speaking communities, namely DENIC (www.denic.de), NIC.AT (www.nic.at), and SWITCH (www.nic.ch) to inform Latin GP's solution regarding the IDNA 2003 compatibility issue.

At the second level, the .DE registry (DENIC) offers 00DF as a separate, stand-alone code point[1]; in consequence these hypothetical domain names "straße.de" and "strasse.de" would be offered for registration as two separate domains[2]. The .CH registry (SWITCH) and the .AT registry (nic.at) do not offer 00DF in their repertoires for the second level per their published policies[3] [4].

Input from the German User Community
The Latin GP has sought input from experts of the three major German-speaking ccTLDs (namely Denic, nic.at, and switch, for Germany, Austria, and Switzerland, respectively) on the topic of whether ß and ss should be considered variants. After some discussions, these experts found the following consensus solution, which they suggested to the GP for use at LGR level:

Table D.3 Solution Suggested by the German User Community

Group	ß vs ss							
Source			Target			Variant Candidate [Yes/No]	Disposition [Allocatable/ Blocked]	Rationale
Code Point	Glyph	Unicode Name	Code Point	Glyph	Unicode Name			
00DF	ß	Latin Small Letter Sharp S	0073 + 0073	ss	Latin Small Letter S + Latin Small Letter S	Yes	Allocatable	See Section 6.6.2
0073 + 0073	ss	Latin Small Letter S + Latin Small Letter S	00DF	ß	Latin Small Letter Sharp S	Yes	Blocked	See Section 6.6.2

The experts from the German-speaking ccTLD of German users suggested two main reasons for creating this variant relation:

[1] DENIC Domain Name Guidelines: https://www.denic.de/fileadmin/public/documents/DENIC_Domainrichtlinien_EN.pdf
[2] https://www.denic.de/en/know-how/idn-domains/
[3] SWITCH IDN Policy: https://www.nic.ch/faqs/idn/
[4] NIC.AT Repertoire: https://www.nic.at/media/files/pdf/IDN-Zeichentabelle.pdf

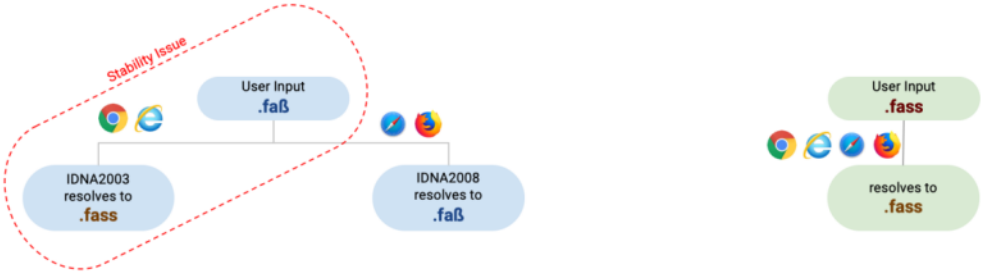
1. There are still browsers (e.g., Chrome) that apply IDNA 2003 at the time of writing. Users of such browsers have each ß automatically replaced by a sequence of two s.
2. Swiss users do not use ß and consider it as equivalent to ss, even where they are able to recognize and point out the differences, when pressed to do so. In consequence, a Swiss user would e.g., very likely rewrite an IDN as .strasse even where it had been presented to the same user .straße before. Therefore, a variant relationship is warranted on non-visual grounds.

For the variant disposition, the same experts were of the opinion that ß needs to be allocatable towards ss, since the same transformation is done by IDNA 2003 and since the same is a long-standing and widely-applied orthographic solution by the German-language community also outside of IDNs, considered valid by all users, especially in the context of domain names. For the other direction, however, the experts were of the opinion that the disposition should be blocked, since there are many non-German words having a double ss (e.g., cross, process, discussion) for which the same label with ß makes no sense (e.g., croß, proceß, discußion), which would lead to the generation of too unintended allocatable variants otherwise.

Possible Solutions to Address the IDNA 2003 Compatibility Issue for Latin Small Letter Sharp S (ß) 00DF: Pros and Cons

Based on the evidence presented, the GP tried to weigh different solutions to address the IDNA 2003 Compatibility issues, which are summarized in Diagram D.3:

Diagram D.3: General Factors to Resolving the IDNA 2003 Compatibility Issue in the Case of Latin Small Letter Sharp S (ß) 00DF



	Option 1	Option 2	Option 3	Option 4
	Exclude ß	**Include ß with allocatable variant to "ss" (ß → ss: a)**	**Include ß with blocked variant to "ss" (ß → ss: b)**	**Include ß without variant to "ss"**
Eliminates stability issue	🟢	🟠	🟠	🔴
Addresses failure of service	🟢	🟠	🔴	🔴
Addresses misconnection	🟢	🟢	🟢	🔴

The pros and cons for each solution are presented in more detail in the following tables:

Table D.3. Solution excluding 00DF from the Latin script repertoire

Pros	Cons
● Most conservative option; removes the option of DNS labels with code point 00DF. The possibility of landing at the "wrong" website is greatly diminished because there would be only one version of the website (i.e. the one using 'ss' (0073 0073)).	● Misconnection or failure of service is still possible when using Chrome or Edge (albeit only one domain name would actually exist) because user input is independent of whether a domain name exists or not. ● Code point 00DF is used in the orthography of German as written in Germany and Austria (but not in Switzerland). German is an EGIDS level 1 language. ● For the German and Austrian part of the user community it would force a fallback for all names and words ordinarily spelled with 00DF. In many such cases, the spelling with "ss" actually identifies a different name or word (minimal pair), which effectively would restrict the linguistic freedom of these communities[5].

Table D.3. Solution including 00DF with a variant relationship with 'ss' (ß → ss)

Pros	Cons
● The possibility of landing at the "wrong" website is diminished provided the two versions of domain names are controlled by the same entity. ● Enables labels that match the actual spelling for certain words and names in the German and Austrian part of the user community; code point	● Limits registration choices. ● Due to transitivity there will be a variant relationship ß (Latin Small Letter Sharp S, 00DF) → 'ss' → β (Greek Beta, 03B2), therefore imposing a cross-script variant on the Greek script LGR. ● Failure of service or misconnection

[5] Cf. also the discussion on pp. 115-117 of [HUSSAIN] summarizing how linguistic rights are recognized politically today as a form of freedom of expression.

00DF is used in the orthography of German as written in Germany and Austria (but not in Switzerland). German is an EGIDS level 1 language.	may occur depending on an application's implementation (IDNA 2003 or IDNA 2008 + TR46). ● Due to overlap of "ss" with 's', there will be additional variants consisting of pairs of all variants of 's' to ensure that variant label sets are well-behaved. See RFC 8228.

Table D.4. Solution for Disposition: Allocatable versus Blocked ß → ss

2.1 ß → ss: Allocatable	**2.2 ß → ss:** Blocked
● It would be possible for a registry operator to apply for the variant label. Per the latest IDN variant TLD Management Framework recommendation, each TLD variant should be evaluated and processed as a standalone TLD (i.e., separate application fee, evaluation process, etc.) ● If registry operator does not apply for the variant label, the label will remain reserved for said registry operator. ● Misconnection cannot occur but a failure of service can. ● Because there are words that contain multiple instances of 00DF, measures must be implemented to prevent more than two allocatable variants, i.e., one for the German/Austrian spelling and one for the Swiss.	● With a "blocked" disposition, the variant label would remain withheld from registration by any registry operator. ● Misconnection cannot occur but a failure of service can.

Table D.5. Solution for Disposition: Allocatable versus Blocked ss → ß

2.3 ss → ß: Allocatable	**2.4 ss → ß:** Blocked
● Simpler solution for TLD applicant; the TLD applicant does not need to be concerned about asymmetrical relationship. Can apply for the 'ss' version first and apply for the 00DF version at a later point in time. ● German-language users do not expect that all labels spelled with double 'ss' can also be represented with a label with Sharp S (00DF); Swiss users expect a label with Sharp S (00DF) to always be represented with a label with double 'ss', while German and Austrian users may accept that as a fallback.	● Alignment with LGR procedure (i.e., minimize allocatable variants) ● No linguistic expectations on the side of the users. ● Most conservative option according to the LGR Procedure ● Denies the opportunity to apply for the 00DF version, if 'ss' is registered first.

Table D.6. Solution to Include 00DF without variant relationship with 'ss'

Pros	**Cons**
● Option is consistent with implementation by DENIC (German registry); German users have been conditioned to this behavior.	● Failure of service or misconnection may occur depending on the application's implementation (IDNA 2003 or IDNA 2008 + TR46) with respect to ß. ● Forces applicants to register all label combinations to defend against spoofing (unless this is robustly excluded as part of the TLD process). In some cases, there are more than two possible combinations.

	• Confusing for Swiss people as they generally use ‘ss’ in all cases for Sharp S (00DF). • Need for defensive registration

Conclusion: Inclusion of 00DF with Variant Mechanism
The Latin GP proposes a solution that balances the needs of certain parts of the Latin script community while minimizing security and stability issues introduced by applications outside the DNS. The solution is to include Latin Small Letter Sharp S (00DF) with a variant relationship with the sequence of letters ‘ss’ (0073 0073), as follows:

Table D.7. Final Variant Solution for Latin Small Letter Sharp S (00DF)

Source Code Point	Variant Relationship	Target Code Point	Disposition
00DF Latin Small Letter Sharp S	→	0073 0073 Latin Small Letter S + Latin Small Letter S	Allocatable
0073 0073 Latin Small Letter S + Latin Small Letter S	→	00DF Latin Small Letter Sharp S	Blocked

This LGR solution along with the appropriate policies (i.e., TLD variant labels managed by the same entity, and second level variant labels managed by the same registrant) would not solve the failure of service problems but would mitigate the issues of misconnection.

D.5.2 Latin Small Letter Dotless I (ı) 0131

There are four Latin code points that have a special case (upper case/lower case) relationship:

- U+0069 Latin Small Letter I ("i")
- U+0049 Latin Capital Letter I ("I")
- U+0131 Latin Small Letter Dotless I ("ı")
- U+0130 Latin Capital Letter I with Dot Above ("İ")

In most system [locale] settings Latin Small Letter I is lower case of Latin Capital Letter I, and reverse Latin Capital Letter I (U+0069) is upper case of Latin Small Letter I (U+0069). In those system locale settings, Latin Capital Letter I (U+0049) is also upper case of Latin Small Letter Dotless I, but not the reverse. It could be described as in the following chart:

Table D.8. Case Relationships for 0069, 0049, 0130, and 0131

Character	Process	Resulting Character	Process	Resulting Character
Latin Small Letter I U+0069	up case →	Latin Capital Letter I U+0049	down case →	Latin Small Letter I U+0069
Latin Small Letter Dotless I U+0131	up case →	Latin Capital Letter I U+0049	down case →	Latin Small Letter I U+0069
Latin Capital Letter I with Dot Above U+0130	down case →	Latin Small Letter I U+0069	up case →	Latin Capital Letter I U+0049

In two system [locale] settings, for Turkish and Azeri language settings, respectively, the case relationship is different. In those two, Latin Small Letter I and Latin Capital Letter I with Dot Above are in mutual upcase/downcase relationship to each other, as well as Latin Small Letter Dotless I and Latin Capital Letter I, which could be described as in the following chart:

Table D.9. Case Relationships in Turkish and Azeri Locales

Character	Process	Resulting Character	Process	Resulting Character
Latin Small Letter I	up case →	Latin Capital Letter I with Dot Above	down case →	Latin Small Letter I
Latin Small Letter Dotless I	up case →	Latin Capital Letter I	down case →	Latin Small Letter Dotless I

If we look at the repertoire of Latin code points for the root zone, as proposed by the Latin Generation Panel, Latin Small Letter I and Latin Small Letter Dotless I are included, whereas the capital letters are excluded. Capital letters are not even valid in IDNA 2008, so the question is, is the case relationship described here a problem or even relevant?

Before [IDNA 2008], there was [IDNA 2003]. Even though IDNA 2003 has been replaced by IDNA 2008 it is still implemented. For example, the web browser Google Chrome, and its derivatives, to date remains IDNA 2003 compliant but not fully IDNA 2008 compliant. In IDNA 2003 there is a pre-process, normalization, of domain names before conversion to Punycode. That normalization includes down casing of Latin characters. For ASCII labels there is already an equivalence between upper case and lower case letters. And this is what users, based on decades of experience, expect to happen.

In an IDNA 2003 compliant web browser "EXÄMPLE" and "EXAMPLE" are equivalent to "exämple" and "example", respectively. In an IDNA 2008 browser "EXAMPLE" must be accepted, but "EXÄMPLE" could be rejected since "Ä" is not valid, but that is not how e.g., Mozilla Firefox and Apple Safari have been designed to handle the problem. They also do down case, as a preprocess, before the formal IDNA 2008 process.

Even though down casing is not part of the formal IDNA 2008 process, one of the IDNA 2008 documents, RFC 5894, states that the user interface of an application, before IDNA 2008 processing, can do normalization. The down casing in IDNA 2008 browsers should probably be seen in that light.

While "TÄT" will probably be down cased to "tät" in the browser, what should the browser do with "TIT"? Depending on the locale that the browser is running in, it may be down cased to either "tit" or "tıt" (dotted or dotless, respectively).

The case shift, in an application, is expected to go only in one direction, from upper case to lower case. When domain names are presented in text, however, it is common that domain names are presented in upper or mixed case. So "ice" might become "Ice" (with regular capital I) or "İce" (with capital I with dot above).

It is quite obvious from the text above that case shift of dotted or dotless I could create erroneous lookup, but the question is how large threat it would be to the users. Since the applications are expected to go from upper case to lower case, when they handle domain names, we should consider a situation where down casing could result in different lower case letters, i.e., when CAPITAL LETTER I is down cased.

With a non-Turkish and non-Azeri system locale setting, a Latin Capital Letter I in a domain name is down cased to Latin Small Letter I in an IDN label. In an ASCII label it could be used as equivalent to Latin Small Letter I or down cased to Latin Small Letter I (both give the same result in DNS).

With a Turkish or Azeri locale, a Latin Capital Letter I is expected to be down cased to Latin Small Letter Dotless I, but in an ASCII label in a domain name, it is still expected to be equivalent with Latin Small Letter I, because that is what the DNS standards says. The Latin GP conducted a small informal study of two web browsers, Firefox and Chrome, in Turkish locale setting. The question was what the web browser would do with a domain name with a non-ascii label

containing a Latin Capital Letter I, e.g. "TÄMPI.example.com". Instead of following the specification of the Turkish locale, the web browsers, in this test, followed the normal DNS behavior. The example would be downcased to "tämpi.example.com".

In spite of the behavior of the web browsers, there is an obvious risk that, in a Turkish or Azeri locale that the two letters are confused or mistreated due to the case folding, and this confusion could be misused. To be on the safe side Latin Small Letter I and Latin Small Letter Dotless I are to be variants.

The Latin GP discusses with Turkish expert from TR ccTLD manager. The TR ccTLD manager recommends that “i” and “ı” should be variants. The Turkish users are used to the habit of using “i” instead of “ı”, therefore the label using “i" is ambiguous with the same label using “ı”.

The example of usage: An IT company ‘ıssız communications’ wants to apply for a TLD ‘ıssız’, while a restaurant ‘issiz restaurant’ wants to apply for a TLD ‘issiz’.

- Both “ıssız” and “issiz” are legitimate words in Turkish. They have different meanings.
- The communications company (ıssız) would be happy to have “issiz” as TLD, too, due to current habit of using “i” instead of “ı” in Turkey.
- The restaurant will not be interested in having “ıssız” as TLD, in addition to “issiz”, since nobody would be expected to use “ı” instead of “i”.

Based on the example, the mapping type from Latin Small Letter Dotless I to Latin Small Letter I is allocatable and the mapping type from Latin Small Letter I to Latin Small Letter Dotless I is blocked.

Table D.10. Variant Relationships for 0069 and 0131

Group	Dotless i vs. i							
Source			Target			Variant Candidate [Yes/No]	Disposition [Allocatable/ Blocked]	Rationale
Code Point	Glyph	Unicode Name	Code Point	Glyph	Unicode Name			
0069	i	Latin Small Letter I	0131	ı	Latin Small Letter Dotless I	Yes	Blocked	No need of allocatable
0131	ı	Latin Small Letter Dotless I	0069	i	Latin Small Letter I	Yes	Allocatable	IDNA 2003 compatibility