Pre-Delegation Testing

IDN Test Cases

Version O

File name: PDT_IDN_TC.docx
Last saved: 2015-05-20

Document control

Document information and security

Made by	Responsible for fact	Responsible for document
Cary Karp	Cary Karp	Cary Karp

Security class	File name
External	PDT_IDN_TC.docx

Revisions

Date	Version	Name	Description
2013-01-13	PA1	Cary Karp	Initial document
2013-02-07	PA2	Cary Karp	Input requirements clarified; Several EPP tests segregated into new TC
2013-02-08	PA3	Rickard Bellgrim	Update text and rearrange test cases
2013-02-11	PA4	Lennart Bonnevier	Review text
2013-03-04	PA5	Cary Karp	Modified in response to editorial review
2012-03-04	PA6	Rickard Bellgrim	Verify input data against the application
2013-04-08	B	Staffan Hagnell	Delivery D2 for production
2013-05-03	C	Mats Dufberg	
2013-06-16	Cbis	Cary Karp	Clarify pass/fail criteria. Migrate all EPP testing into IDNvalid07 and IDNvalid08.
2013-07-04	D	Cary Karp	Add IDNvalid00.
2013-07-10	E	Cary Karp	Substantive editorial consolidation and clarification of ver. D.
2013-07-11	F	Cary Karp	Reworded IDNvalid05. Added requirement for UTF-8 encoding of non-ASCII text.
2013-07-22	G	Cary Karp	Clarification of test detail.
2013-09-19	H	Cary Karp, Mats Dufberg	Harmonized with the IDN section added to the Self-Certification Document.
2014-01-16	I	Mårten Frosth, Mats Dufberg	Harmonized TC with new IDN Self-Certification Document. Updated all TC with simplified descriptions of steps and criteria for NA, PASS and FAIL.
2014-02-10	J	Cary Karp	Addition of test for RFC 5893 RTL contextual rules. Clarification of other test detail.
2014-08-07	K	Mårten Frosth	Update of IDNvalid07 regarding testing of variant activation policy and warning criteria.
2014-10-10	L	Mats Dufberg	Update of IDNvalid04, -05 and -06.
2015-03-16	M	Mats Dufberg	Added test cases IDNvalid09 and -10. Updated IDNvalid07 and IDNvalid03. Moved IDNvalid08 to IDNvalid11.

Date	Version	Name	Description
2015-04-15	N	Mats Dufberg	IDNvalid03 updated on the requirements on Modifier Letters.
2015-05-20	O	Mats Dufberg	IDNvalid03 updated again on the requirements on Modifier Letters.

LIST OF CONTENTS

1. Introduction

1.1 Scope

The Pre-Delegation Testing Provider will verify that there is a one-to-one correspondence between the languages or scripts listed in Exhibit A of the Applicant's Registry Agreement and the IDN tables provided to the PDT; that each table is explicitly associated with a single script or language; that each table is formatted according to RFC 4290 or RFC 3743 or a local format for which ICANN has given dispensation; that each listed code point is valid under the IDNA protocol; that every string of tabulated code points permissible as a registered label conforms to the IDN Guidelines: that all policy and context-dependent requirements of IDNA and the Guidelines are clearly stated and enforced in the registry; that a complete list has been provided of the language or script tags for the IDN EPP Extension, if such is required (e.g. 'fr' for a French language table, or 'cyrl' for a Cyrillic script table); that all policies associated with the management of variant relationships among tabulated code points are documented and enforced; that a specimen EPP transaction has been provided illustrating how to register a new IDN label, including a language or script tag if they are used.

1.2 References

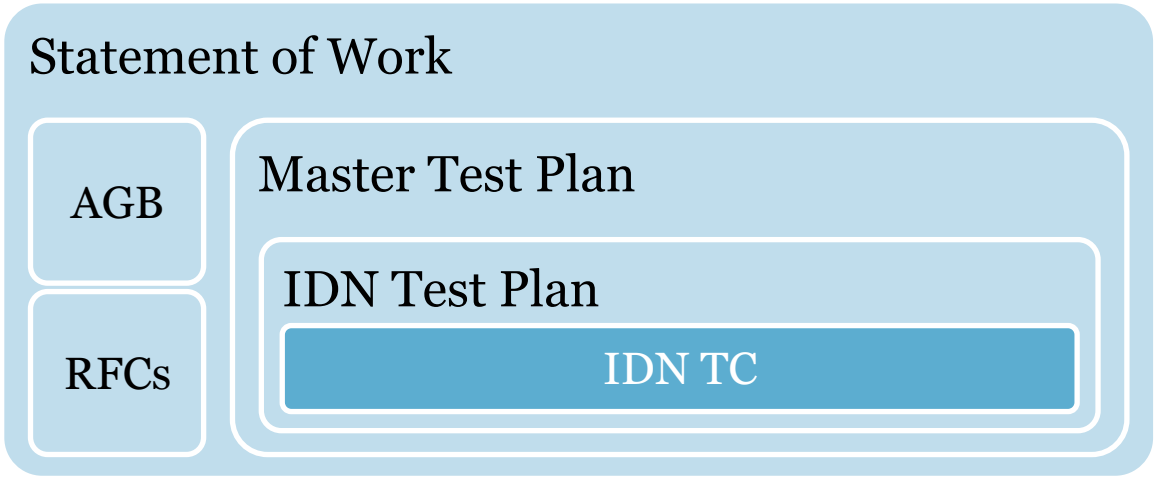
1.2.1 External

- ICANN gTLD Applicant Guidebook, Version 2012-06-04
- ICANN Guidelines for the Implementation of Internationalized Domain Names, Version 3.0 (“IDN Guidelines”)
- IEEE 829-2008
- RFC 3743
- RFC 4290
- RFCs 5890 through 5894 (“IDNA”)
- The Unicode Standard (“Unicode”)
- Registry Agreement for the TLD under test

1.2.2 Internal

- Pre-Delegation Testing, Statement of Work
- Pre-Delegation Testing, Master Test Plan
- Pre-Delegation Testing, IDN Test Plan

1.2.3 Document Hierarchy



1.3 Context

The tests have two basic elements. The first is the offline review of a table and the associated policy statements. The second is determining the registry response to the attempted registration of labels constructed specifically to verify that code points and strings not permitted for registration are rejected, and those that are permitted are accepted, with particular attention to contextual constraints imposed in the reference documents and the additional normative instruments they invoke. The online test components are aggregated into a single test, IDNvalid11.

The tests are supported algorithmically to the extent possible. However, variation in the tabulation of nominally identical code point repertoires and the substance and format of the associated policy statements, necessitates a significant amount of manual testing.

Some tests require the generation of test labels. This will initially be part of the case-by-case action of the test officers but will gradually result in a library of generally applicable test labels. In any further case where a test label needs to be custom designed, it will be added to that repository for reuse where appropriate in subsequent testing.

1.4 Notation for description

Each IDN test case is described under a separate heading, below. The test procedures are described with the test case to which they apply.

2. IDNvalid00

2.1 Test case identifier

IDNvalid00 - Verification of submitted tables, EPP extensions, and policy statement.

2.2 Objective

This test verifies that the IDN tables and documents listed in Exhibit A of the Applicant's Registry Agreement have all been submitted to PDT for testing and that no submitted tables or documents are unlisted in the Registry Agreement.

2.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TableList	A list of all script and language tables cited in, the Exhibit A of the Registry Agreement.	File, Registry Agreement.
PolicyStatement	The IDN policies declared by the registry in section 4 of the IDN Self-Certification Document, corresponding to TableList.	File, IDN Self-Certification Document.
EPPtags	A list of all IDN EPP extensions, as well as language and script tags, needed for the registration of an IDN label.	File
TestTable	The table under scrutiny.	File or files.

2.4 Outcome(s)

The response to this test will be a pass/fail/non-applicable determination. If this test case ends with "not applicable", then no further test cases will be performed.

2.5 Environmental needs

- Basic desktop.

2.6 Special procedural requirements

None.

2.7 Intercase dependencies

None.

2.8 Ordered description of steps to be taken to execute the test case

1. Determine whether the registry is authorized to support IDNs based on Exhibit A of the Registry Agreement, and if it is not, abort the remaining sequence of IDN tests.
2. Verify that every submitted TestTable corresponds to an item in TableList and that every item in TableList corresponds to a submitted TestTable.

3. Verify that the PolicyStatement in section 4 of the IDN Self-Certification Document correlates to TestTable(s) in a manner that unambiguously indicates:
    a. How requests for registration of IDN labels will be processed.
    b. How the Registry handles comingling of scripts.
    c. How the Registry handles variants.
    d. How the Registry handles contextual rules.
4. For the IDN tables listed in the response to Section 1 of the IDN Self-Certification Document, verify that the corresponding tables are listed in Section 3 of that document, that every element of EPPtags corresponds to a specific TestTable and that there are no orphaned extensions or tables.
5. Verify that all IDN tables have been submitted as TXT files and that any which include non-ASCII text are encoded in Unicode UTF-8.

Criteria for NA:

- Exhibit A of the Registry Agreement does not declare support for IDN labels (Step 1).

Criteria for PASS:

- Each script or language listed in Exhibit A corresponds to a submitted table and no submitted table is unlisted (Step 2),
- PolicyStatement unambiguiously indicates how IDN labels are processed (Step 3a-3d),
- all IDN tables are listed in the IDN Self-Certification Document Section 1 and 3; that all elements of EPPtags correspond to specific IDN tables; that there are no orphaned extentions or tables (Step 4),
- all IDN tables are submitted as TXT in UTF-8 (Step 5).

Criteria for FAIL:

- The conditions in Steps 2-5 are not met or if part of the information is unclear or missing.

If this test fails, further testing will be suspended pending remedial action. If this is not undertaken within the prescribed time, the failure will be confirmed, none of the subsequent tests will be conducted, and all will fail by default.

3. IDNvalid01

3.1 Test case identifier

IDNvalid01 - IDN table validation.

3.2 Objective

This test verifies that the format of a code point table either conforms to RFC 4290 or RFC 3743, or is in an adequately documented alternative format. The test is repeated for all tables.

3.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The table under scrutiny.	File
LocalTableFormat	Describes the table format in Section 2 of the IDN Self-Certification Document if it does not comply with either of the reference RFCs.	File, IDN Self-Certification Document
LocalTableJustification	Verifiable warrant in Section 2 of the IDN Self-Certification Document for using a local format instead of either of the reference RFCs.	File, IDN Self-Certification Document

Neither of the reference RFCs specifies a rigorous enough format for TestTable to be automatically parsed for conformance, and there is no way to predict the details of an instance of LocalTableFormat. Manual examination of a table is necessary in order to validate the format.

3.4 Outcome(s)

The response to this test will be a pass/fail/warn determination.

3.5 Environmental needs

- Basic desktop.

3.6 Special procedural requirements

The person running this test must understand the elements of an IDN table format, both as described in the reference RFCs, and in order to assess the sufficiency of a locally defined alternative and the justification for its use.

3.7 Intercase dependencies

None.

3.8 Ordered description of steps to be taken to execute the test case

1. Verify that the response to Section 2 of the IDN Self-Certification Document indicates whether the table format used follows the guidelines of RFC 4290, RFC 3743 or if

justification has been included in Section 2 of that document explaining why neither could be used.

2. If the response to Section 2 of the IDN Self-Certification Document indicates that the table is in RFC 4290 format verify that:
    a. each code point in the table appears in Unicode U+nnnn notation.
    b. if the base character has any variants, the indication of its code point is followed by a VERTICAL LINE.
    c. if the base character has more than one variant, the code points for the variants are separated by a COLON.
    d. if the base character has a variant composed of a sequence of characters they are indicated with a HYPHEN MINUS between each code point.
    e. comment lines in the table are preceded with a NUMBER SIGN.
3. If the response to Section 2 of the IDN Self-Certification Document indicates that the table is in RFC 3743 format verify that:
    a. each code point in the table appears without the Unicode U+ prefix , or if the general exception permitting the the use of the Unicode "U+" prefix is invoked, that each code point in the table appears in correct U+nnnn notation.
    b. if the valid code point has any variants, the columns are separated by a SEMICOLON.
    c. if there are multiple preferred or character variants, they are separated by a COMMA.
    d. if a variant is composed of a sequence of code points they are separated by a SPACE.
    e. if references are indicated, the reference number is listed in PARENTHESIS directly after the code point it and that the source is included in the list in the beginning of the IDN table.
    f. comments in the table are preceded with a NUMBER SIGN.
    g. that the version number and release date are indicated.
4. If the response to Section 2 of the IDN Self-Certification Document indicates that neither of the reference RFC table formats was used, verify that a justification has been included in Section 2 of that document as to why. Verify that LocalTableFormat supports the functionality necessary to conduct the other tests.

Criteria for PASS:

- Section 2 of the IDN Self-Certification Document indicates the table format used for the IDN table (Step 1 and 4),
- if the table is in RFC 4290 format, that it conforms to those guidelines (Step 2a-2e),
- if the table is in RFC 3743 format, that it conforms to those guidelines (Step 3a-3g),
- if the table format follows neither the guidelines in RFC 3743 nor RFC 4290, the way in which the alternative is to be understood must be unambiguously apparent (Step 4).

Criteria for FAIL:

- Section 2 of the IDN Self-Certification Document does not indicate the selected table format was used,
- the conditions in Step 2-4 are not met,
- part of the information is unclear or missing.

If this test fails, none of the subsequent tests will be conducted and all will fail by default.

4. IDNvalid02

4.1 Test case identifier

IDNvalid02 - IDNA code point validation.

4.2 Objective

This test verifies that the status of each tabulated code point is PROTOCOL VALID (PVALID) or CONTEXTUAL RULE REQUIRED (CONTEXTn) as defined in RFC 5892 when its algorithms are applied to the Unicode Standard, version 6.3. The test is repeated for all tables.

The IDNA Derived Property (PVALID, CONTEXTO, CONTEXTJ, DISALLOWED or UNASSIGNED) can be found for each code point on http://www.iana.org/assignments/idna-tables-6.3.0/idna-tables-6.3.0.xhtml.

4.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	See Section 2.3, above.	File
ExtendedTestTable	Table generated by test IDNvalid02.	File
AvailableCodepointTable	A tabular listing of all PVALID and CONTEXTn code points in Unicode 6.3, with separate columns indicating the IDN status and the Unicode script property value for the code point that keys every row. This file is provided internally.	File

4.4 Outcome(s)

The output will be an extended version of TestTable with new columns added for IDN status and Unicode script property values. This will be assigned the ID ExtendedTestTable and used as input for subsequent tests. There will also be a pass/fail/warn determination.

4.5 Environmental needs

- Basic desktop.
- Text sorting and comparison utilities.

4.6 Special procedural requirements

None.

4.7 Intercase dependencies

IDNvalid01.

4.8 Ordered description of steps to be taken to execute the test case

For every row in TestTable, keyed on the first code point appearing in it, determine if there is a corresponding row in AvailableCodepointTable, and if there is, generate ExtendedTestTable and end the test as passed. If any row is keyed with a code point that does not also key a row in AvailableCodepointTable, end the test as failed.

1. Examine ExtendedTestTable, and verify that each code point has one of the three derived IDNA property values PVALID, CONTEXTJ, or CONTEXTO.

Criteria for PASS:

- Each code point in ExtendedTestTable has one of the three derived IDNA property values PVALID, CONTEXTJ, or CONTEXTO (Step 1).

Criteria for FAIL:

- ExtendedTestTable includes one or more code points indicating IDNA property values other than PVALID, CONTEXTJ or CONTEXTO (Step 1).

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary. If this test fails, none of the subsequent tests will be conducted and all will fail by default.

5. IDNvalid03

5.1 Test case identifier

IDNvalid03 - Context Rule Validation.

5.2 Objective

This test verifies that a tabulated code point that require contextual rules can only be used according to those rules. The test is repeated for all IDN tables. This concerns:

1. Code points with the IDN property CONTEXTJ or CONTEXTO can only be used according to the contextual rule given in RFC 5892.
2. The contextual prohibitions on mixing Arabic and European digits in right-to-left labels, and on digits at the start of such labels given in RFC 5893.
3. Combining marks (nonspace, spacing or enclosing marks) can never start a label (RFC 5891).
4. Required contextual rules for Modifier Letter. More on Modifier Letters below.
5. Most COMMON and INHERITED code points are usually restricted to be used with certain explicit Unicode scripts (UAX#24 and ScriptExtensions.txt of the Unicode database).
6. Code points with restricted context (e.g. must not be initial) according to Unicode.
7. All strings must be in Normalization Form C (see http://www.unicode.org/reports/tr15/ and http://www.unicode.org/Public/6.3.0/ucd/).

Each Modifier Letter (General Unicode category value Modifier_Letter, Lm) must be accompanied with contextual rule restricting it to relevant positions or be accompanied with a statement that no such rule is needed for that Modifier Letter. If ICANN has published requirements on a certain Modifier Letter, those must be followed. Stricter rules are permitted. Each rule or non-rule should be justified with documentation showing the Modifier Letters use in the language in question (for language based IDN tables) or some language using that script (for script based IDN tables). If ICANN has published requirements or recommendation on contextual rules for a certain Modifier Letter, it is valid to refer to that document.

For some Unicode scripts and many languages IDN Reference Tables are available in Github at <https://github.com/dotse/IDN-ref-tables>. The IDN Reference Tables contain relevant contextual rules and documentation on Modifier Letter, if available.

5.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
ExtendedTestTable	Table generated by test IDNvalid02.	File
IDNAContextualRules	The contextual rules listed in RFC 5892, Appendix A.	File, RFC 5892.
BidiDigitRules	The contextual rules given in RFC 5893, Section 2.	File, RFC 5893.
UnicodeContextRules	The contextual rules given in Unicode	Unicode
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-

		Certification Document.

5.4 Outcome(s)

The response to this test will be a pass/fail/non-applicable/warn determination.

5.5 Environmental needs

- Basic desktop.

5.6 Special procedural requirements

The person conducting this test must understand the application of the CONTEXTn rules in RFC 5892, the Bidi rule in RFC 5893, and Unicode.

5.7 Intercase dependencies

This test effectively extends into IDNvalid11.

5.8 Ordered description of steps to be taken to execute the test case

1. Examine ExtendedTestTable. If the IDN property CONTEXTO or CONTEXTJ does not appear on any row in it, the table does not contain code points that can form RTL labels, and the table contains no other code points requiring contextual rules, end this test as non-applicable.
2. For every code point with the IDN properties CONTEXTO or CONTEXTJ, verify that the availability is restricted as required by RFC 5892 (see Appendix A).
3. If the label is RTL as defined in RFC 5893:
    a. if it contains any code point in the range 0030..0039, ensure that no code point is in the range 0660..0669, and vice versa,
    b. if the literal component of the label consists of code points taken with the explicit script property value Arabic, ensure that no ARABIC DIGIT, EXTENDED ARABIC DIGIT, or European DIGIT is in the initial position.
4. For every code point in the General Unicode category Combining mark, verify that the availabilty is restricted to non-initial position.
5. For every code point in the General Unicode category Modifier Letter, verify that the availability is restricted by relevant contextual rule or accompanied with a statement that no rule is required.
6. For every code point in the General Unicode category Modifier Letter verify that supporting documentation of such use is provided; or if no contextual rule is given it is shown that unrestricted placement is permissible for that code point through supporting documentation.
7. For every code point with special property COMMON or INHERITED (except for HYPHEN-MINUS U+002D and ASCII digits 0030..0039) verify that it has the restrictions that the Unicode Standard prescribes, if any.
8. Verify that there exist contextual rules, if needed, to prevent any string to be registrered unless it is in Normalization Form C.

For every code point besides those mentioned above, verify if the Unicode standard prescribes any restrictions, and if so, that those are followed (e.g. U+0E40).
Criteria for NA:

- The IDN properties CONTEXTO or CONTEXTJ do not appear in ExtendedTestTable, nor do any other code points that require contextual rules.

Criteria for PASS:

- The IDN property CONTEXTO or CONTEXTJ appears in ExtendedTestTable (Step 2) and the code points to which they are assigned are restricted to use as required by RFC 5892 (Step 2a-2i) and this is indicated unambiguously in PolicyStatement.
- The restrictions on digits in RTL labels required by RFC 5893 are observed (step 3).
- Other code points requiring contextual rules have appropriate rules (steps 4-6)

Criteria for WARN

- The criteria for PASS are fullfilled except that the justification for the contextual rule(s) or absence of contextual rule(s) for any Modifier Letter is missing, incomplete or unable to be validated by the testing provider. The TC will end with a warning accompanied with a comment explaining the warning. A warning is a not a failure condition.

Criteria for FAIL:

- The IDN property CONTEXTO or CONTEXTJ appears in ExtendedTestTable but the required contextual restraints are not indicated in PolicyStatement (Step 2a-2i).
- ARABIC-INDIC digits and European digits appear together in an RTL label.
- An RTL label begins with an ARABIC-INDIC DIGIT, or an EXTENDED ARABIC-INDIC DIGIT, or a European DIGIT.
- Combining mark appears in the table, but without contextual rules restricting it from initial position.
- Modifier Letter appears in the table, without contextual rule restricting its use or without statement that no contextual rules are needed.
- Modifier Letter appears in the table that ICANN has published with a requirement for contextual rules on that Modifier Letter and the provided contextual rule does not meet the requirement.
- Code points with property COMMON or INHERITED appear in the table without contextual rules that the Unicode standard requires.
- The code point repertoire and rules make it possible to construct a string not in Normalization Form C that would be accepted.
- Other code point without proper restriction.

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary. If this test fails, none of the subsequent tests will be conducted and all will fail by default.

6. IDNvalid04

6.1 Test case identifier

IDNvalid04 - IDN script validation.

6.2 Objective

This test verifies that the code point array in a script table is restricted to a single explicit Unicode script property value as defined in the Unicode Standard Annex #24, that code points with the special script property values COMMON or INHERITED are correctly associated with the designated script, and that regardless of script property value, no code point is used in a manner alien to the designated script. The test is repeated for all tables.

For some Unicode scripts IDN Reference Tables are available in Github at <https://github.com/dotse/IDN-ref-tables>. When such a table is available matching the ExtendedTestTable in question it is consulted to determine if the code points in the ExtendedTestTable are consistent with the Unicode script. If the ExtendedTestTable contains some code points beyond those in the relevant reference table, those code points will be considered before this TC is run based on the evidence of use provided. See Appendix B in this document for suggestions of evidence for additional code points. If accepted by the PDT Service Provider, the reference table will be updated.

6.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
ExtendedTestTable	Table generated by test IDNvalid02.	File
AvailableCodepointTable	A tabular listing of all PVALID and CONTEXTn code points in Unicode 6.3, with separate columns indicating the IDN status and the Unicode script property value for the code point that keys every row. This file is provided internally.	File
ScriptIntegrityPolicies	The script integrity policies declared by the registry.	File
UAX#24	Unicode Standard Annex #24; Unicode Script Property.	File

6.4 Outcome(s)

The response to this test will be a pass/fail/warning determination.

6.5 Environmental needs

- Basic desktop.

6.6 Special procedural requirements

The person conducting this test must understand Unicode script properties designating specific scripts, as well as the values COMMON and INHERITED. These are described in UAX #24, which states that COMMON and INHERITED are assigned to code points that are used with more than one script but that this does not imply usability with all scripts. UAX #24 does not provide unequivocal guidance on how to apply such restrictions but does illustrate correct and incorrect use of those properties.

The underlying principles are to be applied in a contextually appropriate manner. For the purpose of the IDN level of the PDT this is taken to mean that any script identifier appearing in the Unicode character name given to a COMMON or INHERITED code point must be congruent with the identifier of the IDN table being tested. For example, a Cyrillic script table may not include the code point named ARABIC FATHATAN, nor would that code point be permissible in a Danish language table.

Similar constraints apply to combining marks and modfier letters. Regardless of their script property values, these may not be randomly interspersed in a string. The appearance, for example, of U+0483 (COMBINING CYRILLIC TITLO) must be restricted to contexts where it is appropriate, which do not include U+047D (CYRILLIC SMALL LETTER OMEGA WITH TITLO).

The only code points with the COMMON script property that may be accepted in any IDN table are 0030..0039 DIGIT ZERO..DIGIT NINE, and U+002D HYPEN-MINUS. This is the digit and hyphen component of the basic ASCII LDH repertoire and will be referred to as "DH" in the following text. The full LDH repertoire (DH plus 0061..007A) will also be accepted if a script table is primarily based on CJK Unified Ideographs or Hangul Syllables.

Any other use of COMMON or INHERITED code points in a language table will require justification as being necessary to support the established orthographic practice of that language.

A script-based test table that indiscriminately includes all COMMON and INHERITED code points will fail.

6.7 Intercase dependencies

The outcome of this test may be contingent upon IDNvalid05. It also effectively extends into IDNvalid11.

6.8 Ordered description of steps to be taken to execute the test case

1. Examine ExtendedTestTable. If the column indicating the script property value contains the same explicit script property value for every row in the table and the table is labeled as supporting the designated script, end the test with pass.
2. If the special script property value COMMON appears in the table and the value INHERITED does not, verify that every COMMON code point is in the DH cluster.
3. If the explicit script property value is Han, Hangul, Hiragana or Katakana, and Latin code points are included in the table, verify that they are in range 0061..007A and that IDNvalid05 is pass.

4. If a code point that is not in the DH cluster has the value COMMON or if a code points has the value INHERITED, verify that the conditions discussed in UAX#24 are met.

Criteria for PASS:

- ExtendedTestTable indicates the same explicit Unicode script property value for every listed code point and the table is correctly labeled as supporting that script (Step 1).
- If the special script property value COMMON appears in a table and the value INHERITED does not, every code point is in the DH cluster (Step 2).
- If the explicit script property value is Han, Hangul, Hiragana or Katakana, and Latin code points in the range 0061..007A are included in the table, IDNvalid05 is pass (Step 3).
- If the Unicode script property values COMMON or INHERITED appear, the conditions discussed in Section 6.6, above, are met (Step 4).

Criteria for FAIL:

- The Unicode script property values COMMON or INHERITED appear, and the conditions discussed in Section 6.6, above, are not met (Step 4).

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary.

7. IDNvalid05

7.1 Test case identifier

IDNvalid05 - IDN script-mixing rule validation.

7.2 Objective

This test verifies that a table including code points with more than one script property value is associated with rules that enforce the constraints on script mixing specified in the IDN Guidelines. The test is repeated for all tables.

7.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
ExtendedTestTable	Table generated by test IDNvalid02.	File
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.

7.4 Outcome(s)

The response to this test will be a pass/fail/warning determination.

7.5 Environmental needs

- Basic desktop.

7.6 Special procedural requirements

None.

7.7 Intercase dependencies

This test may determine the outcome of IDNvalid04. It also effectively extends into IDNvalid11.

7.8 Ordered description of steps to be taken to execute the test case

1. Examine ExtendedTestTable. If the column indicating the Unicode script property contains only one explicit script designator and no COMMON or INHERITED code points, end this test with pass.
2. If that column contains one explicit script property value and COMMON or INHERITED code points are present, verify that they are appropriate to that script.
3. If that column contains more than one explicit script property value, verify that one of the following conditions is met:
    a. The mixing of scripts in a table is restricted to LDH code points with Hangul Syllabes.
    b. The mixing of scripts in a table is restricted to LDH code points with Unified CJK Ideographs.

c. The mixing of scripts in a table is restricted to LDH code points with Unified CJK Ideographs intermingled with Hiragana or Katakana.
d. PolicyStatement in Section 4 of the IDN Self-Certification Document explain and justify the conditions under which the intermingling of the indicated scripts is permitted.

Criteria for PASS:

- The column indicating the Unicode script property of ExtendedTestTable contains only one explicit script property value and no COMMON or INHERITED code points.
- The column indicating the Unicode script property of ExtendedTestTable contains only one explicit script property value and all listed COMMON or INHERITED code points are appropriate to that script (Step 2), or,
- The column indicating the Unicode script property of ExtendedTestTable contains more than one explicit script property value and one of the conditions in Step 3a-3d is met.

Criteria for FAIL:

- The column indicating the Unicode script property of ExtendedTestTable contains only one explicit script property value but COMMON or INHERITED code points are incongruous with the explicitly designated script (Step 2).
- The column indicating the Unicode script property of ExtendedTestTable contains more than one explicit script property value and none of the conditions in Step 3a-3d is met.

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary.

8. IDNvalid06

8.1 Test case identifier

IDNvalid06 - IDN language validation.

8.2 Objective

This test verifies that a table associated with a language rather than a script is consistent with the script-based constraints in the preceding test cases, and that linguistic warrant is demonstrated in any policy statement permitting the intermingled use of multiple scripts in individual labels. The test is repeated for all tables.

For many languages IDN Reference Tables are available in Github at <https://github.com/dotse/IDN-ref-tables>. When such a table is available for the languages in question it is consulted to determine if the code points in the TestTable are consistent with the language use. If the TestTable contains code points beyond those in the relevant reference table, those code points will be considered before this TC is run based on the evidence of use provided. See Appendix B in this document for suggestions of evidence for additional code points. If accepted by the PDT Service Provider, the reference table will be updated.

8.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The table under scrutiny.	File
ExtendedTestTable	Table generated by test IDNvalid02.	File
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.
IDN Reference Table	IDN table found at https://github.com/dotse/IDN-ref-tables	

8.4 Outcome(s)

The response to this test will be a pass/fail/non-applicable/warn determination.

8.5 Environmental needs

- Basic desktop.

8.6 Special procedural requirements

The person conducting this test must be familiar with basic concepts of writing systems and have access to reference material about the code point repertoires associated with the languages figuring in the PDT. Special care is needed in situations where a language uses multiple scripts but only one of them appears in a label. For example, although the Japanese writing system includes both the Latin and Katakana scripts, in a label consisting exclusively of Latin code points, U+30FC (KATAKANA-HIRAGANA PROLONGED SOUND MARK) would not be permissible.

8.7 Intercase dependencies

None.

8.8 Ordered description of steps to be taken to execute the test case

1. If TestTable is labeled as supporting a script rather than a language, end this test as non-applicable.
2. If TestTable supports a language, examine ExtendedTestTable and verify that:
    a. one explicit script property value is indicated and it is appropriate to the writing system for the designated language, and that the supported repertoire is used for that writing system. If available, an IDN Reference Table will be used to evaluate the ExtendedTestTable.
    b. more than one explicit script property value is indicated and the table is declared to support a language with a writing system that uses all of those scripts, that PolicyStatement in Section 4 of the IDN Self-Certification Document provides verifiable warrant for that assertion.

Criteria for NA:

- If TestTable is labeled as supporting a script rather than a language, end this test as non-applicable (Step 1).

Criteria for PASS:

- The code point repertoire in a language table is appropriate to the writing system of the indicated language (Step 2a). Broad allowance will be made for documentable orthographic variation.
- The code point repertoire in a language table is declared to support a language with a writing system that uses multiple scripts and PolicyStatement in Section 4 of the IDN Self-Certification Document provides verifiable warrant for that assertion (Step 2b.

Criteria for FAIL:

- The code point repertoire in a language table is not appropriate to the writing system of the indicated language (Step 2a) or the indiscriminate inclusion of additional code points from the script(s) used for that writing system.
- The code point repertoire in a language table is declared to support language with a writing system that uses multiple scripts and PolicyStatement in Section 4 of the IDN Self-Certification Document does not provide verifiable warrant for that assertion (Step 2b).

NOTE: the only writing system thus far figuring in the discussion of IDN repertoires that uses multiple scripts is Japanese, which intermingles elements of the Han, Hiragana, Katakana, and the Basic Latin scripts ("a..z"). As noted in IDNvalid04, the PDT also accepts the Basic Latin repertoire together with Unified CJK Ideographs or Hangul Syllables without need for separate justification.

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary.

9. IDNvalid07

9.1 Test case identifier

IDNvalid07 - IDN variant code point validation.

9.2 Objective

This test verifies that policies for the processing of variant relationships between listed code points are described in sufficient detail, and that all code points listed in a submitted table as having variant relationships are concordant with those policies. The test is repeated for all tables.

9.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The table under scrutiny.	File
VariantAlgorithms	The IDN policies submitted for IDNvalid00 that describe the variant generation algorithms used by the registry.	File, IDN Self-Certification Document
VariantPolicies	The IDN policies submitted for IDNvalid00 that describe the variant management policies declared by the registry.	File, IDN Self-Certification Document
GRsupport	The yes/no response to the question in Section 1 of the applicant's IDN Self-Certification Document regarding support for IDN at the start of General Registration.	File, IDN Self-Certification Document
EPPtags	A list of all EPP extensions needed to submit a request for the registration of an IDN label.	File

9.4 Outcome(s)

The response to this test will be a pass/fail/non-applicable/warning determination.

9.5 Environmental needs

- Basic desktop.
- EPP client.
- IPv4 or IPv6 connectivity.

9.6 Special procedural requirements

The person conducting this test must understand the concept of variant code points that ICANN applies to IDN repertoires and the associated registration policies. Further procedural constraints are discussed in Section 2.5.2 of the IDN Test Plan. Special care needs to be taken with scripts that have single-code point and multiple-code point representations of the same character (i.e. both precomposed and combining forms, without the one canonically being

replaced by the other). If both forms are included in a table, variant policies must be provided to enusre that they cannot be separately delegated.

9.7 Intercase dependencies

None.

9.8 Ordered description of steps to be taken to execute the test case

1. Examine TestTable. If every row in it includes only one code point, and no row indicates any correlation between its code point and a code point on any other row, end this test as non-applicable.
2. If correlations between two code points are indicated ("variant relationships"), verify that
    a. VariantPolicies in Section 4 of the IDN Self-Certification Document explains each such relationship and the constraints that attach to it, or VariantAlgorithms in Section 4 of the IDN Self-Certification Document describes the processing of each such relationship. If neither is available, end the test as failed.
    b. VariantPolicies in Section 4 of the IDN Self-Certification Document describe how the Registry Operator activates variants and that it correlates with what is stated in Exhibit A of the applicant's Registry Agreement regarding the activation of variants. If there are any discrepancies, issue a warning to the applicant.
3. If GRsupport is negative, the remaining steps in this test are omitted. If GRsupport is positive, proceed with the test sequence but restrict it to the tables that are explicitly listed Section 1 of the IDN Self-Certification Document and are also listed in Exhibit A of the applicant's Registry Agreement.
4. If there is any uncertainty about how variant management policies are applied to a table in a regard that is significant to the pass/fail determination, construct a label including a code point that is expected to be replaced by another code point upon registration and submit an EPP request for it. If the request is accepted without any indication of that transformation having been applied, end the test as failed.
5. If there is similar uncertainty about the variant management process blocking the registration of a label in one variant form if a label in another variant form has already been registered, construct a second test label in a form that should cause such blocking, and submit an EPP request for it. A further test label in a form that is not expected to be blocked may also be submitted for registration.

This EPP component of this test is only applicable to tables that are listed both in Exhibit A of the applicant's Registry Agreement *and* Section 1 of their IDN Self-Certification Document.

Criteria for NA:

- This test is not applicable to tables that do not declare variant relationships between listed code points and where no such relationships are otherwise apparent (Step 1).

Criteria for PASS:

- Any table that indicates variant relationships between code points must be accompanied by documentation that clearly explains how those relationships are managed in the registry (Step 2).

- If EPP tests are conducted, the registry must accept EPP requests for the registration of labels so that the behavior expected on the basis of the documentation can be verified. EPP extensions that are required in this process is included in the documentation.

- If the registry accepts and rejects test labels in accordance with the anticipated behavior, or the documentation of variant management is sufficient without EPP testing (Step 4 and 5).

Criteria for FAIL:

- The Section 4 of the IDN Self-Certification Document does not explain the variant management in sufficient detail to support the test (Step 2).
- Required EPP extensions are not included in the documentation.
- Expected EPP responses are not returned (Step 4 and 5).

10. IDNvalid08

10.1 Test case identifier

IDNvalid08 - The test case has been moved to IDNvalid11 to maintain the logical order of the test cases.

11. IDNvalid09

11.1 Test case identifier

IDNvalid09 - Variant management.

11.2 Objective

This test verifies that the variant management as described in the policy is compliant with the regulation of variant management in the Registry Agreement for that TLD.

11.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.
ExhibitA	Exhibit A of the Registry Agreement between ICANN and the Registry Operator for the TLD tested.	File

11.4 Outcome(s)

The response to this test will be a pass/fail/not applicable determination.

11.5 Environmental needs

- Basic desktop.
- EPP client.
- IPv4 or IPv6 connectivity.

11.6 Special procedural requirements

None.

11.7 Intercase dependencies

IDNvalid07.

11.8 Ordered description of steps to be taken to execute the test case

1. If IDNvalid07 has resulted in "not applicable" then end with NA.
2. Examine PolicyStatement and determine what the policy for variant management is.
3. Examine ExhibitA and determine if the TLD is permitted to have any variant management.
4. Examine ExhibitA and determine if the TLD is permitted to activate any variants and what rules there are for activation.
5. Compare the results of step 3 and 4 with the result of step 2.
6. If the variant management in the PolicyStatement is compliant with the permissible variant management, end with PASS.

7. If the variant management in the PolicyStatement is NOT compliant with the permissible variant management, end with FAIL.

Criteria for NA:

- IDNvalid07 is NA (Step 1).

Criteria for PASS:

- PolicyStatement is compliant with ExhibitA (Step 6).

Criteria for FAIL:

- PolicyStatement is compliant with ExhibitA (Step 7).

12. IDNvalid10

12.1 Test case identifier

IDNvalid10 - Basic IDN compliance.

12.2 Objective

This test verifies that the registry system is compliant with basic IDN requirements by not accepting the registration of neither IDN strings in U-label format with HYPHEN in position three and four nor non-IDN ASCII strings with HYPHEN in position three and four. It also verifies that an IDN string in U-label format is rejected if it has HYPHEN in initial or final position.

12.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.
GRsupport	The yes/no response to the question in Section 1 of the applicant's Self-Certification Document regarding support for IDN at the start of General Registration (see also IDNvalid08).	File, IDN Self-Certification Document
EPPtags	A list of all IDN EPP extensions, such as language and script tags, needed to submit a request for the registration of an IDN label (see also IDNvalid08).	File

12.4 Outcome(s)

The response to this test will be a pass/fail.

12.5 Environmental needs

- Basic desktop.
- EPP client.
- IPv4 or IPv6 connectivity.

12.6 Special procedural requirements

None.

12.7 Intercase dependencies

IDNvalid11.

12.8 Ordered description of steps to be taken to execute the test case

1. Inspect PolicyStatement to determine if the TLD accepts registration of ASCII domains without any EPP/IDN extension. Unless it explicitly says that registration of ASCII domains are not supported, assume that it is.
2. The following test strings are to be registered without any EPPtag.
    a. Construct TL91 as a unique, valid ASCII label and "expect accept". TL91 is skipped if ASCII domains are not registerable.
    b. Construct TL92 as a unique, otherwise valid non-IDN ASCII label, but with HYPHEN in third and forth positions and set "expect reject". TL92 is always included.
3. If GRsupport is negative, skip the following steps.
4. If GRsupport is positive, select one of the the tables that are explicitly listed in Section 1 of the IDN Self-Certification Document.
5. Determine the EPPtag used for the table, if any.
6. The following test strings are to be registered as other IDN labels in IDNvalid08.
    a. Construct TL93 as a unique, otherwise valid IDN label matching all other restrictions of the selected table, but with HYPHEN in third and fourth positions of the U-label.
    b. Construct TL94 as a unique, otherwise valid IDN label matching all other restrictions of the selected table, but with HYPHEN in the initial positions of the U-label.
    c. Construct TL95 as a unique, otherwise valid IDN label matching all other restrictions of the selected table, but with HYPHEN in the final positions of the U-label.
7. Submit a request to register TL91 to TL95. This is done together with the test labels from IDNvalid11, if applicable.

Criteria for PASS:

- Expected result is returned. All the following applies (only constructed labels are considered):
    a. TL91 is accepted.
    b. TL92 is rejected.
    c. TL93 is rejected.
    d. TL94 is rejected.
    e. TL95 is rejected.

Criteria for FAIL:

- Expected result is not returned. Any of the following applies (only constructed labels are considered):
    a. TL91 is rejected.
    b. TL92 is accepted.
    c. TL93 is accepted.
    d. TL94 is accepted.
    e. TL95 is accepted.

13. IDNvalid11

13.1 Test case identifier

IDNvalid11 - IDN online registry response verification.

13.2 Objective

This test verifies that the online registry correctly processes test strings needed for preceding tests. The test is repeated for all tables.

13.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
ExtendedTestTable	Table generated by test IDNvalid02.	File
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.
GRsupport	The yes/no response to the question in Section 1 of the applicant's Self-Certification Document regarding support for IDN at the start of General Registration.	File, IDN Self-Certification Document
EPPtags	A list of all IDN EPP extensions, such as language and script tags, needed to submit a request for the registration of an IDN label.	File

13.4 Outcome(s)

The response to this test will be a pass/fail/warning determination.

13.5 Environmental needs

- Basic desktop.
- EPP client.
- IPv4 or IPv6 connectivity.

13.6 Special procedural requirements

None.

13.7 Intercase dependencies

This test is an effective extension of IDNvalid03, IDNvalid04, and IDNvalid05. It is, however, only applied to those tables listed in Exhibit A of the applicant's Registry Agreement that are also listed in their response to Section 1 of the corresponding IDN Self-Certification Document. If no tables are listed in that section, this test is not applicable.

Test labels constructed in this test case are registred together with the test labels constructed in test case IDNvalid10.

13.8 Ordered description of steps to be taken to execute the test case

1. If GRsupport is negative, terminate this test as not applicable. If GRsupport is positive, proceed with the test sequence but restrict it to the tables that are explicitly listed in Section 1 of the IDN Self-Certification Document and are also listed in Exhibit A of the applicant's Registry Agreement.
2. Examine ExtendedTestTable, for every code point or group of code points that the execution of IDNvalid03 has shown to have contextual rules, construct three test labels including the code point with that property.
    a. The first test label, TL1 will place the code point in the context required by the associated rule.
    b. The second, TL2, will place the code point in a context that violates the rule.
    c. The third, TL3, will include two instances of the code point, of which one will respect the contextual rule and the other will violate it, if applicable.
3. Examine ExtendedTestTable and construct three test labels or sets of labels:
    a. The first, TL4 consists solely of listed code points.
    b. The second, TL5, includes one code point that is not listed.
    c. The third, TL6, includes at least one code point with an explicit script property value that both differs from any listed in the table, and is not allowed in PolicyStatement.
4. Examine ExtendedTestTable and if it contains code points in both the range 0030..0039 and the range 0660..0669, and is an RTL label according to RFC 5893, construct a test label:
    a. that includes code points from both ranges, TL7.
5. Examine ExtendedTest Table, and if it includes code points from any Arabic script block construct three test labels:
    a. The first, TL8, begins with a code point in the range 0030..0039 and is followed by code points with the explicit script property value Arabic.
    b. The second, TL9, begins with a code point in the range 0660..0699 and is followed by code points with the explicit script property value Arabic.
    c. The third, TL10, begins with a code point in the range 06F0..06F9 and is followed by code points with the explicit script property value Arabic.
6. Submit a request to register (use any elements of EPPtags that may be necessary) TL1 through TL10. Include applicable teststrings from IDNvalid10.

This test is only applicable to tables that are listed both in Exhibit A of the applicant's Registry Agreement *and* Section 1 of their IDN Self-Certification Document. Any EPP extensions required for the submission of a registration request must be included in the documentation.

Criteria for NA:

- GRsupport is negative (Step 1).

Criteria for PASS:

- EPP extensions required for the submission of a registration request is included in the documentation.
- Expected result is returned (Step 4).
    a. TL1 is accepted.
    b. TL2 is rejected.
    c. TL3 is rejected.
    d. TL4 is accepted.
    e. TL5 is rejected.
    f. TL6 is rejected.
    g. TL7 is rejected.
    h. TL8 is rejected.
    i. TL9 is rejected.
    j. TL10 is rejected.
- The expected EPP result code is returned for each test label derived from a preceding test case.

Criteria for FAIL:

- EPP extensions required for the submission of a registration request are not included in the documentation.
- Expected result is not returned (Step 4).
    a. TL1 is rejected.
    b. TL2 is accepted.
    c. TL3 is accepted.
    d. TL4 is rejected.
    e. TL5 is accepted.
    f. TL6 is accepted.
    g. TL7 is accepted.
    h. TL8 is accepted.
    i. TL9 is accepted.
    j. TL10 is accepted.
- The expected EPP result code is not returned for each test label derived from a preceding test case.

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary or if the response to the request indicates that IDN registration is not yet supported in the registry.

14. Global

14.1 Glossary

The glossary is available in the Master Test Plan.

14.2 Document change procedures

Document change procedures are documented in the Master Test Plan.

15. Appendix A: CONTEXTO and CONTEXTJ code points

A code point with the IDN properties CONTEXTO or CONTEXTJ is restricted as required by RFC 5892:

- U+200C (ZERO WIDTH NON-JOINER). This may occur in a formally cursive script (such as Arabic) in a context where it breaks a cursive connection as required for orthographic rules, for example, in the Persian language. It may also occur in Indic scripts in a consonant-conjunct context (immediately following a VIRAMA), to control required display of such conjuncts.
- U+200D (ZERO WIDTH JOINER). This may occur in Indic scripts in a consonant-conjunct context (immediately following a VIRAMA), to control required display of such conjuncts.
- U+00B7 (MIDDLE DOT) is used to permit the Catalan character ela geminada to be expressed and must be preceded and followed by a LATIN SMALL LETTER L ("l" U+006C).
- U+0375 (GREEK LOWER NUMERAL SIGN, KERAIA) is only permitted in Greek script.
- U+05F3 (HEBREW PUNCTUATION GERESH) must be preceded by a Hebrew script code point.
- U+05F4 (HEBREW PUNCTUATION GERSHAYIM) must be preceded by a Hebrew script code point.
- U+30FB (KATAKANA MIDDLE DOT) is only permitted in a label if at least one other code point in the label is Hiragana, Katakana or Han script.
- 0660..0669 (ARABIC-INDIC DIGITS) cannot be mixed with EXTENDED ARABIC-INDIC DIGITS.
- 06F0..06F9 (EXTENDED ARABIC-INDIC DIGITS) cannot be mixed with ARABIC-INDIC DIGITS.

16. Appendix B. Suggestions for evidence

The following are non-exhaustive examples of justification for a code point inclusion in a language table:

- A generally recognized authority for the language in question accepts the character associated with the code point.
- Published dictionaries list the character associated with the code point as belonging to the written tradition of the language in question.
- The character associated with the code point is commonly used by publishing industry (books or newspapers) for words in that language.
- Education material (e.g. "abc" books or equivalent) use the character assiciated with the code point.

The following are the criteria for a code point inclusion in a script table:

- The code point has the script property in question and is compliant with IDNA 2008.
- The code point has the script property COMMON or INHERITED, is recognized by the Unicode Standard for that explicit script property and is compliant with IDNA 2008.