

Pre-Delegation Testing

IDN Test Cases

Version F

File name: PDT_IDN_TC.docx

Last saved: 2013-07-11

Copyright (c) 2013 Internet Corporation For Assigned Names and Numbers. All rights reserved.

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.

LIST OF CONTENTS

1. INTRODUCTION	5
1.1 SCOPE	5
1.2 REFERENCES	5
1.2.1 External	5
1.2.2 Internal	5
1.2.3 Document Hierarchy	5
1.3 CONTEXT.....	6
1.4 NOTATION FOR DESCRIPTION	6
2. IDNVALIDoo	7
2.1 TEST CASE IDENTIFIER	7
2.2 OBJECTIVE	7
2.3 INPUTS	7
2.4 OUTCOME(S)	7
2.5 ENVIRONMENTAL NEEDS	7
2.6 SPECIAL PROCEDURAL REQUIREMENTS.....	7
2.7 INTERCASE DEPENDENCIES	7
2.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	7
3. IDNVALIDo1	9
3.1 TEST CASE IDENTIFIER	9
3.2 OBJECTIVE	9
3.3 INPUTS	9
3.4 OUTCOME(S)	9
3.5 ENVIRONMENTAL NEEDS	9
3.6 SPECIAL PROCEDURAL REQUIREMENTS.....	9
3.7 INTERCASE DEPENDENCIES	9
3.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	10
4. IDNVALIDo2.....	11
4.1 TEST CASE IDENTIFIER	11
4.2 OBJECTIVE	11
4.3 INPUTS	11
4.4 OUTCOME(S)	11
4.5 ENVIRONMENTAL NEEDS	11
4.6 SPECIAL PROCEDURAL REQUIREMENTS.....	11
4.7 INTERCASE DEPENDENCIES	11
4.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	11
5. IDNVALIDo3.....	13
5.1 TEST CASE IDENTIFIER	13
5.2 OBJECTIVE	13
5.3 INPUTS	13
5.4 OUTCOME(S)	13
5.5 ENVIRONMENTAL NEEDS	13
5.6 SPECIAL PROCEDURAL REQUIREMENTS.....	13
5.7 INTERCASE DEPENDENCIES	13
5.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	13
6. IDNVALIDo4.....	15
6.1 TEST CASE IDENTIFIER	15
6.2 OBJECTIVE	15
6.3 INPUTS	15
6.4 OUTCOME(S)	15
6.5 ENVIRONMENTAL NEEDS	15
6.6 SPECIAL PROCEDURAL REQUIREMENTS.....	15

6.7	INTERCASE DEPENDENCIES	16
6.8	ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	16
7.	IDNVALIDo5.....	17
7.1	TEST CASE IDENTIFIER	17
7.2	OBJECTIVE	17
7.3	INPUTS	17
7.4	OUTCOME(S)	17
7.5	ENVIRONMENTAL NEEDS	17
7.6	SPECIAL PROCEDURAL REQUIREMENTS.....	17
7.7	INTERCASE DEPENDENCIES	17
7.8	ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	17
8.	IDNVALIDo6.....	19
8.1	TEST CASE IDENTIFIER	19
8.2	OBJECTIVE	19
8.3	INPUTS	19
8.4	OUTCOME(S)	19
8.5	ENVIRONMENTAL NEEDS	19
8.6	SPECIAL PROCEDURAL REQUIREMENTS.....	19
8.7	INTERCASE DEPENDENCIES	19
8.8	ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	19
9.	IDNVALIDo7.....	21
9.1	TEST CASE IDENTIFIER	21
9.2	OBJECTIVE	21
9.3	INPUTS	21
9.4	OUTCOME(S)	21
9.5	ENVIRONMENTAL NEEDS	21
9.6	SPECIAL PROCEDURAL REQUIREMENTS.....	21
9.7	INTERCASE DEPENDENCIES	21
9.8	ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	21
10.	IDNVALIDo8.....	23
10.1	TEST CASE IDENTIFIER	23
10.2	OBJECTIVE	23
10.3	INPUTS	23
10.4	OUTCOME(S)	23
10.5	ENVIRONMENTAL NEEDS	23
10.6	SPECIAL PROCEDURAL REQUIREMENTS.....	23
10.7	INTERCASE DEPENDENCIES	23
10.8	ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE	23
11.	GLOBAL	25
11.1	GLOSSARY	25
11.2	DOCUMENT CHANGE PROCEDURES	25

1. Introduction

1.1 Scope

The Pre-Delegation Testing Provider will verify that each IDN 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 that can be registered as a 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 all policies associated with the management of variant relationships among tabulated code points are similarly documented and enforced.

1.2 References

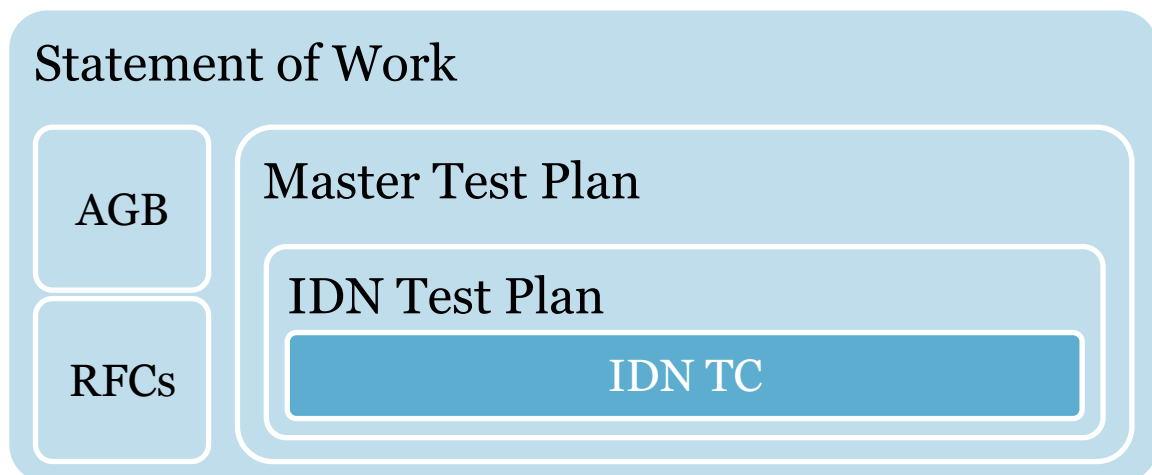
1.2.1 External

- 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")

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, IDNvalid08.

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 statements.

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 for testing and that no submitted tables or documents are unlisted in the 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
PolicyStatement	The statement of IDN policies cited in Exhibit A of the Registry Agreement.	File
EPPtags	A list of all EPP extension values, such as language tags, needed to submit a request for the registration of an IDN label.	File
TestTable	The table under scrutiny.	File

2.4 Outcome(s)

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

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. Verify that every submitted TestTable corresponds to an item in TableList and that every item in TableList corresponds to a submitted TestTable.
2. Verify that PolicyStatement correlates to TestTable(s) in a manner that unambiguously indicates how requests for the registration of IDN labels will be processed, that no policy detail required to understand a table is missing, and that every policy is supported by the table to which it refers.

3. Verify that every element of EPPtags corresponds to a specific TestTable and that there are no orphaned extensions or tables.
4. Verify that all documents and tables have been submitted as TXT files and that any which include non-ASCII text are encoded in Unicode UTF-8.

This and all subsequent tests are non-applicable if Exhibit A of the Registry Agreement does not declare support for IDN labels. This test will pass if each script or language listed in Exhibit A corresponds to a submitted table, no submitted table is unlisted, and all non-ASCII text is encoded in Unicode UTF-8. This test will fail if these conditions are not met or if there is an imbalance between the submitted tables and requisite EPP tags.

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 4920 or RFC 3743, or is in an acceptable adequately documented alternative format.

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 if it does not comply with either of the reference RFCs.	File
LocalTableJustification	Verifiable warrant for using a local format instead of either of the reference RFCs.	File

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 any case, to determine whether relevant policy statements are interposed with the structured code point listings as narrative commentary. The validation of format compliance can easily be conducted during the course of that examination, leaving no reason for segregating the ordered component of the table for subsequent automated processing.

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. Compare the TestTable format with the one prescribed in RFC 4920 and if it conforms end the test as passed.
2. Compare the TestTable format with the one prescribed in RFC 3743 and if it conforms end the test as passed.
3. If LocalTableJustification adequately explains why neither of the reference RFC formats could be used, and LocalTableFormat supports the functionality necessary to conduct the other tests, end the test as passed. If these criteria are not met, end the test as failed.

This test will pass if two conditions are fulfilled. First, TestTable must conform to one of the two prescribed reference formats, or if it does not, the way in which the alternative is to be understood must be unambiguously apparent. Second, TestTable must be parsable by the script at <https://github.com/dotse/idn-properties>. This test will fail if either of these conditions is not met. 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.

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 6.2.

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
AvailableCodepointTable	A tabular listing of all PVALID and CONTEXTn code points in Unicode 6.2, 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

- 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.

This test will pass if each code point in ExtendedTestTable has one of the three derived IDNA property values PVALID, CONTEXTJ, or CONTEXTO. This test will fail if any other value is

indicated. 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 - IDNA Context Rule Validation

5.2 Objective

This test verifies that a tabulated code point with the status CONTEXTJ or CONTEXTO can only be used according to the contextual rule for that code point given in RFC 5892.

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

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 the reference RFC.

5.7 Intercase dependencies

This test effectively extends into IDNvalid08 with the test labels TL1, TL2, and TL3.

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

1. Examine ExtendedTestTable
2. If the IDN property CONTEXTO or CONTEXTJ does not appear on any row in it, terminate this test as non-applicable.
3. If these properties appear, determine if all code points to which they are assigned are restricted to use as required by RFC 5892.
4. If all code points conform, end the test as passed.
5. If they do not conform, end the test as failed.

This test will pass if it is clear from the table and documentation that IDNAContextualRules are observed for every code point with the values CONTEXTJ or CONTEXTO. This test will fail if either of these properties is present without application of the required contextual constraints. If

no code points have these properties, this test is not applicable. A warning will be issued if failure is not directly indicated but a qualifying remark is necessary.

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, and that code points with the special script property values COMMON or INHERITED are correctly associated with the designated script.

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 6.2, 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.

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 HYPHEN-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

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 test table that indiscriminately accepts all code points with COMMON and INHERITED script properties for use with all scripts will fail.

6.7 Intercase dependencies

This test effectively extends into IDNvalid08.

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

1. Examine ExtendedTestTable.
2. 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.
3. If the special script property value COMMON appears in the table and the value INHERITED does not, if every COMMON code point is in the DH cluster, end the test as passed.
4. If a code point that is not in the DH cluster has the value COMMON or there are code points with the value INHERITED, and the conditions discussed in UAX#24 are disregarded, end the test as failed.

This test will pass if the following conditions are met. If ExtendedTestTable indicates the same explicit Unicode script property value for every listed code point and the table is correctly labeled as supporting that script, the test will pass. If the Unicode script property values COMMON or INHERITED appear, the conditions discussed in Section 6.6, above, must also be met for the test to pass. In all other cases, the test will fail. 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.

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
ScriptIntegrityPolicies	The script integrity policies declared by the registry.	File

7.4 Outcome(s)

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

7.5 Environmental needs

- Basic desktop

7.6 Special procedural requirements

None.

7.7 Intercase dependencies

This test effectively extends into IDNvalid08.

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

1. Examine ExtendedTestTable
2. If the column indicating the Unicode script property contains only one explicit script designator, end the test as non-applicable.
3. If that column contains more than one explicit script property value, determine if ScriptIntegrityPolicies explain and justify the conditions under which the comingled use of the indicated scripts is permitted. If there is no such statement, end the test as failed.
4. If there is a satisfactory such statement end the test as passed.

This test will pass if all code points in a table have the same explicit Unicode script property value and all listed COMMON or INHERITED code points are appropriate to that script. This test will fail if no justification is provided for the comingled use of multiple script property values or

if COMMON or INHERITED code points are incongruous with the explicitly designated script. This test is not applicable if only one explicit script property value appears and there are no special script property values. 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.

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
LanguagePolicies	The language support policies declared by the registry.	File

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.

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.
3. If only one explicit script property value is indicated, verify that it is appropriate to the writing system for the designated language and that the supported repertoire is used for that writing system.
4. If the column indicating script property value contains more than one explicit script designator and the table is declared to support a language with a writing system that uses

all of those scripts, determine that LanguagePolicies provides verifiable warrant for that assertion.

5. If the two preceding points are fulfilled, end the test as passed.
6. If they are not, end the test as failed.

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. The PDT also accepts the appearance of the Basic Latin repertoire in other CJK contexts without need for separate justification.

This test is not applicable to tables that are identified by script. This test will pass if the code point repertoire in a language table has a reasonable relationship to the writing system of the indicated language. Broad allowance will be made for documentable orthographic variation but the indiscriminate inclusion of additional code points from the script(s) used for that writing system will result in failure. 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.

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 variant generation algorithms used by the registry.	File
VariantPolicies	The variant management policies declared by the registry.	File
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

- 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.

9.7 Intercase dependencies

None.

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

1. Examine TestTable
2. 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.
3. If correlations between two code points are indicated ("variant relationships"), verify either that VariantPolicies explains each such relationship and the constraints that attach

to it, or that VariantAlgorithms describes the processing of each such relationship. If neither is available, end the test as failed.

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, the behavior of the registry may be tested by constructing a label including a code point that is expected to be replaced by another code point upon registration and submitting an EPP request for it. If the request is accepted without any indication of that transformation having been applied, end the test as failed. If the response to the EPP request indicates that IDN registration is not yet supported in the registry, end the test with a warning.
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, a second test label may be constructed in a form that should cause such blocking, and submitted for registration. A further test label in a form that is not expected to be blocked may also be submitted for registration.
6. If the registry accepts and rejects test labels in accordance with the anticipated behavior, or if the documentation of variant management is sufficient without EPP testing, end the test as passed.
7. If the documentation is inadequate or expected EPP responses are not returned, end the test as failed.

This test is not applicable to tables that do not indicate variant relationships between listed code points. This test will pass if the following conditions are met. 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. 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. Any EPP extensions that are required in this process must be included in the documentation. The test will result in a warning if the response to an EPP request indicates that IDN registration is not yet supported in the registry.

The test will fail if the documentation does not explain the variant management in sufficient detail to support the test, or if any EPP test returns an unexpected result code. A warning will be issued if failure is not directly indicated but a qualifying remark is necessary. A warning will also be issued if the response to an EPP request indicates that IDN registration is not yet supported in the registry.

10. IDNvalid08

10.1 Test case identifier

IDNvalid08 - IDN online registry response verification

10.2 Objective

This test verifies that test strings needed for preceding tests are correctly processed by the online registry.

10.3 Inputs

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

Id	Description	Type
ExtendedTestTable	Table generated by test IDNvalid02.	File
ScriptIntegrityPolicies	The script integrity policies declared by the registry.	File
EPPTags	A list of all EPP extensions needed to submit a request for the registration of an IDN label.	File

10.4 Outcome(s)

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

10.5 Environmental needs

- EPP client
- IPv4 or IPv6 connectivity

10.6 Special procedural requirements

None.

10.7 Intercase dependencies

This test is an effective extension of IDNvalid03, IDNvalid04, and IDNvalid05.

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

1. Examine ExtendedTestTable.
2. For every row in it that indicates the IDN property CONTEXT0 or CONTEXTJ, construct three test labels including the code point with that property. The first test label, TL1 will place the code point in the context required by the associated rule. The second, TL2, will place the code point in a context that violates the rule. 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.
3. Submit a request to register TL1. Use any elements of EPPTags that may be necessary in this and all following steps.

4. If the request is rejected, end the test as failed. If the response to the request indicates that IDN registration is not yet supported in the registry, end the test with a warning.
5. Submit a request to register TL2. If it is accepted, end the test as failed.
6. Submit a request to register TL3. If it is accepted, end the test as failed.
7. Construct a test label consisting solely of listed code points, TL4, and submit a request to register it. If it is rejected, end the test as failed.
8. Construct a test label that includes at least one code point that is not listed, TL5, and submit a request to register it. If it is accepted, end the test as failed.
9. Construct a test label that 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 ScriptIntegrityPolicies, TL6. Submit a request to register it. If it is accepted, end the test as failed.
10. If none of the preceding sub-tests has failed, end the test as passed.

This test will pass if the expected EPP result code is returned for each of the test labels described in Section 9.8, below. Any EPP extensions required for submission of a registration request must be included in the documentation. The test will fail if an unexpected result code is returned. A warning will be issued if failure is not directly indicated but a qualifying remark is necessary. A warning will also be issued if the response to an EPP request indicates that IDN registration is not yet supported in the registry.

11. Global

11.1 Glossary

The glossary is available in the Master Test Plan.

11.2 Document change procedures

Document change procedures are documented in the Master Test Plan.