String Similarity Evaluation Guidelines for the New gTLD Program: 2026 Round

Version: 1.0
Date: 23 July 2026

---

1. Background and Introduction

The String Similarity Evaluation (SSE) is a part of the New gTLD Program: 2026 Round application evaluation, as recommended in the Generic Names Supporting Organization (GNSO) Final Report on the New gTLD Subsequent Procedures Policy Development Process (Section 24). The objective of this evaluation is to prevent user confusion and loss of confidence in the Domain Name System (DNS) resulting from delegation of similar strings. The details of

SSE can be found in the New gTLD Program: 2026 Round Applicant Guidebook (AGB), Section 7.10.

The Phase 1 Final Report on the Internationalized Domain Names Expedited Policy Development Process (IDN EPDP Phase 1 Final Report) adds further details on this process and the scope of string similarity analysis to cover strings and their variant strings. The IDN EPDP Phase 1 Final Report notes that the String Similarity Evaluation Panel (SSE panel) may decide whether and what blocked variant labels to omit when conducting a comparison based on the guidelines and/or criteria that justify such an omission on the basis of a manifestly low level of confusability between the scripts of labels being compared and that the guidelines and/or criteria must be developed for use by the SSE panel (Recommendations 4.2 and 4.3).

Draft String Similarity Review Guidelines were initially published for Public Comment in February 2024. Following these guidelines, ICANN has developed the SSE tool and SSE data. This document provides the updated version of these guidelines.

The comparisons between strings are performed for visual similarity. The target users are expected to be native users of the script and to be reasonably attentive while reading the domain names. With variant strings included in the scope, the number of comparisons increases significantly. Therefore, the String Similarity Evaluation Tool (SSE tool) was developed to perform pre-screening and generate potential contention sets as an input for the SSE panel to analyze and make decisions. The SSE tool uses the String Similarity Evaluation Data (SSE data) developed with the help of script experts, finalized after a Public Comment period. The SSE panel can either use or override the suggestions by the SSE tool and SSE data by providing appropriate rationale.

Following the publication of SSE data, and incorporating input from the initial public comment, the String Similarity Evaluation Guidelines (SSE guidelines) were developed to explain how the SSE panel should make its determination using the SSE tool and the SSE data, and published for Public Comment in November 2025. It is the responsibility of the SSE panel to make the final decision and determination on the String Similarity Evaluation for all applied-for strings.

The SSE identifies if any applied-for gTLD string is visually similar to any existing TLDs or Blocked Names or to any other applied-for TLD strings. The other string evaluations, e.g., Singular/Pural Notification Evaluation, are not in the scope of SSE. For more details, please refer to 2026 Round Applicant Journey.

2. Scope of the String Similarity Evaluation

2.1 SSE Panel Evaluation Process

String Similarity Evaluation involves a preliminary comparison of each applied-for string and its variant strings to corresponding strings and variant strings in relevant categories (See AGB, Section 7.10).

The SSE panel is responsible for reviewing all applied-for strings within the scope defined in the AGB. The SSE tool generates a pre-screening report of potential contention sets, which the SSE panel will take as one of the inputs. The SSE panel may consider additional analysis and inputs as needed.

The SSE panel will adjust, add, or remove potential contention sets in the report generated from the SSE tool as part of their final decision, providing relevant rationale for each decision. It is the responsibility of the SSE panel to make the final decision and determination on contention sets as part of the String Similarity Evaluation.

2.2 Considerations for SSE Evaluation

The SSE data includes the following scope of visual similarity analysis for each script conducted by the respective script experts.

A. Compare all elements[1] to ASCII characters, in uppercase and lowercase form (a-z, A-Z).
B. Compare all elements to each other within the script.
C. Compare all elements to elements in the repertoire of other related scripts (if any).
D. Compare all elements to:
    a. capitalized elements of the related scripts (if applicable).
    b. conjuncts or compounded characters (if applicable for the script).
    c. simple shapes in ASCII, Latin, and other related scripts (e.g., circles “o”, lines “l”, curves “c”, “s”, “u”, “ᑎ”).
    d. underlined elements, as may be used in URL links.

The comparison decision should be based on commonly used fonts and font sizes for that script for on-screen reading or user interface elements, such as Internet browser address bars.

When comparing a string with another string within or across scripts, generally all characters in the string should be either in all lowercase form or all uppercase form, not in mixed form. The comparisons between strings can be between lowercase and lowercase forms, uppercase and uppercase forms, or lowercase and uppercase forms, for example:

- The Latin lowercase “com” (U+0063 U+006F U+006D) and
  the Latin lowercase “corn” (U+0063 U+006F U+0072 U+006E).

- The Latin uppercase “USO” (U+0055 U+0053 U+004F) and
  the Armenian uppercase “USO” (U+054D U+054F U+0555).

- The Latin uppercase “COM” (U+0043 U+004F U+004D) and
  the Cyrillic lowercase “сом” (U+0441 U+043E U+043C).

[1] An element in this context refers to the visual representation of either a single code point or any valid sequence of code points within the repertoire of the script in RZ-LGR reviewed as a single item within a domain label for the purposes of visual similarity evaluation.

One of the exceptions to consider is when only the first letter is in uppercase and the rest of the string is lowercase. The similarity caused by randomly mixing uppercase and lowercase within a string needs to have a strong rationale to be within scope. In some cases, the variant and the casing could create similarity between strings (see Section 5. Additional Significant Considerations of the SSE data).

If the string is in a script which has conjunct characters, the visual similarity analysis should consider both contexts where conjunct characters are properly rendered and, if applicable, cases where they may not be properly rendered. This means they will show up as individual characters due to lack of font or rendering support on a platform. For example, these Devanagari script code points U+0915 (क) U+094D (◌्) U+0937 (ष) are supposed to be rendered in conjunct form as “क्ष”. If the conjunct form cannot be rendered, they may be displayed as separated glyphs as “क ◌्ष”.

If the string is in a script with compound characters, the visual similarity analysis should consider visual similarity of the compound characters and corresponding sequence of characters (e.g. æ and ae).

When comparing a string which includes simple-shape characters, as the simple shapes may exist in multiple scripts, the visual similarity analysis must consider cross-script similarities. For example “ဝဝo” (Myanmar) and “uoo” (Latin), or “vou” (Latin) and “υου” (Greek).

An underline is automatically added via the linkification process by many applications, e.g., example.com. This underlining can potentially cause visual similarity across otherwise distinct strings especially if the difference in two strings is only the potential confusable character at the same position. For examples, ASCII strings “good” and “qood” (“g̲ood” and “q̲ood”) or Armenian strings “gհu” (U+0581 U+056B U+057D) and “qհu” (U+0566 U+056B U+057D) (“gհu” and “q̲hu”).

The similarity due to underlining can be considered in the following three cases.

Case 1: When the underlying completely strikes over or covers the visually distinctive feature from one string and makes it visually identical to the other string.

> For example, the Latin script strings “door” (U+0064 U+006F U+006F U+0072) and “doo̲r” (U+0064 U+006F U+006F U+0331 U+0072) may become visually similar due to underlining: “d̲o̲o̲r̲” and “door”. Similarly, the Latin script strings “ṣol” (U+1E63 U+006F U+006C) and “sol” (U+0073 U+006F U+006C) may become visually similar due to underlining “s̲o̲l̲” and “sol”.

Case 2: When the underlying partially covers the visually distinctive feature from a string and makes it visually similar to the other string.

For example, the Latin script strings “poc” (U+0070 U+006F U+0063) and “ρος” (U+03C1 U+03BF U+03C2); “poc” and “ρος”.

Case 3: When the underlining breaks at a letter or its diacritic which extends to the underlining below. This behavior is not consistent across all scripts and platforms and may require explicit formatting instructions, e.g. using Cascading Style Sheets (CSS) (which may not be provided in general) and so should be investigated on a case-to-case basis.

For example, the effects of the CSS set up using the text-decoration-skip-ink is shown below:

CSS: text-decoration-skip-ink	Displayed text
text-decoration-skip-ink: auto;	door
text-decoration-skip-ink: none;	door

In general, the first underlining case has stronger visual string similarity implications than the second case, which in turn has more similarity than the third case (only when skip ink feature for underlining is reliably available and enabled). A case to case determination needs to be made.

Due to technical limitations, some of the similarity data is omitted from the SSE data files after consultation with the script experts. Such code points have been identified in the SSE data and require close manual review by the SSE panel when appearing in a string. For the list of such code points and additional details, see SSE data, 4.1.3. Neo Brahmi scripts and 4.1.4. Thai script sections.

There are other features which contribute to the visual similarity such as repeated characters, or reordered rendering. If certain characters are repeated, especially when done many times in different strings, then these strings may be harder to differentiate visually, e.g., “atta” vs “attta” is more distinguishable than “attttta” vs “atttttta”. In addition, in some scripts the rendering of certain combining characters may change the visual order, e.g., Myanmar string formed by character U+101D (ဝ) followed by the character U+1031 (ေ) is rendered as “ေဝ” instead of “ဝေ”. Such reordering is also done in other scripts with combining marks.

The SSE panel should look at the above factors discussed in this section across different commonly used fonts, rendering systems, applications, and platforms while making its determination. Similarity considerations should be based on cases which are commonly occurring and not based on isolated situations.

3. Levels of the String Similarity in SSE Data

3.1 Category and Similarity Level

The SSE data consist of the similarity categorization done by script experts between code points, consisting of the following five levels:

1. Identical or near identical (homograph), or already defined variants
2. Highly confusable (strongly similar, or near homograph)
3. Similar
4. Distantly similar (weakly similar)
5. Distinct, not similar

The characters considered the “same” by the script community are defined as variants in the Root Zone Label Generation Rules (RZ-LGR). Such cases are included as part of the Level 1.

Level 1, Level 2, and Level 3 are generally considered visually similar. Level 4 is considered not visually similar and level 5 is for visually distinct cases.

These determinations are made on characters or a sequence of characters (referred to as elements earlier). However, the visual similarity analysis is eventually done on whole strings by the SSE panel. So the SSE panel may decide to override these determinations by providing appropriate rationale as discussed later in this document.

3.2 Scripts and Similarity Level

Visual similarity cannot be considered independent of the script of the intended string. The following script pairs show a particularly strong overlap in letter shapes or they are part of a group of related scripts. They are listed here in descending order of cross-script variants defined in the RZ-LGR. Any counts given below reflect visual variant definitions, so any similarity cases would be in addition to these counts.

- Chinese/Japanese/Korean (80)
- Kannada/Telugu (34)
- Latin/Cyrillic (28)
- Devanagari/Gurmukhi (26)
- Latin/Greek (22)
- Armenian/Latin (19)
- Armenian/Greek (12)
- Armenian/Cyrillic (5)

The following scripts are weakly correlated and have relatively small sets of shapes that are visually correlated; some of them may bear an intrinsic relation, while others may be accidental.

- Han/Katakana and Hiragana (8 – some shapes have common origin, or simply reflect a typographical relationship)

- Tamil/Malayalam (6)
- Katakana/Hiragana (3 – some shapes have common origins)
- Hangul/Han (3 – some Hangul characters reflect accidental typographical similarities with Han)
- Devanagari/Bengali (2)

The following scripts have a more distant or no relation, but happen to share some similar letter shapes. These are usually instances of shapes with low complexity and therefore few distinct characteristics that would allow identifying a letter as belonging to a different script.

- Hebrew/Latin and Greek (1 – accidental)
- Myanmar/Georgian (1 – accidental)
- Malayalam/Latin (2 – simple shapes)
- Myanmar/Latin (2 – simple shapes)
- Oriya/Malayalam (1 – simple shape)
- Malayalam/Latin (1 – simple shape)
- Armenian/Ethiopic (0 – but several accidental matches with uppercase Armenian)
- Latin/Ethiopic (0 – but several accidental matches with uppercase Latin)
- Khmer/Lao/Thai (0 - while strongly related in structure and origin, typically have distinct font styles, making it easy to recognize which script is used)

Lastly, the following scripts don't have any cross-script variant mappings defined in the RZ-LGR:

- Arabic
- Ethiopic
- Gujarati
- Khmer
- Lao
- Sinhala
- Thaana
- Thai

The more distant scripts or those with no cross-script variant characters do not imply that there is no possibility for visual similarity. For example Khmer/Lao/Thai may have visually less similar writing systems but have similar underlying characters. However, the above analysis provides some guidance on similarity across scripts. If the SSE panel concludes that some of these scripts share no similarity, the panel may decide to omit their blocked variant strings when conducting the similarity evaluation.

4. Function and Limitations of SSE Data and SSE Tool

The SSE data and SSE tool provide the following functionalities to assist the SSE process:

1. The SSE data includes visually similar characters within script and across scripts as well as other relevant visual similarity cases due to uppercase, underlining, simple shape

confusion, and ASCII confusion, as discussed in Scope of the String Similarity Evaluation section above.
2. Based on the input from the script experts, the SSE data implicitly omits the cases of low level of confusability between the scripts, e.g. Chinese (Han) and Arabic scripts.
3. The SSE tool can handle the large volume of comparisons introduced by variant strings. Based on the given list of strings, the SSE tool automatically creates their variant strings and compares them.
4. For strings which have a very large number of variant strings, e.g., more than 10,000 variant strings, it will not be feasible for the SSE tool to conduct a complete comparison of all variant strings. In such a case, the SSE tool will only analyze strings and their variant strings up to the limit (e.g., equal to 10,000) set by the administrator. The SSE tool will also identify such strings for the SSE panel to do a detailed manual review.
5. The SSE tool can efficiently provide reports of potential similarity sets by using the SSE data along with the string similarity level calculations.
6. The comparison by the SSE tool includes potential within-script and cross-script visual similarities noted in SSE data.
7. The comparison by the SSE tool uses SSE data developed by script experts who have taken into account all the considerations in the Scope of the String Similarity Evaluation above.

The SSE data and the SSE tool are designed to assist the SSE panel in the string similarity evaluation process. However the final decision on the string similarity is a responsibility of the SSE panel. The limitations of the SSE data and SSE tool include:

1. For the SSE data, the script experts do visual similarity analysis for characters, whereas the SSE panel has to do visual similarity analysis for strings. Because the two are different in scope, the two analyses may diverge in some cases.
2. There might be some differences in interpretation of visual similarity by script experts in the SSE data and by the SSE panel due to the subjective nature of the criteria, especially for the borderline cases.
3. Han script includes 19,865 code points in the Chinese RZ-LGR, therefore, the Chinese script expert utilized machine learning and artificial intelligence methods to identify and categorize code point pairs before reviewing and making final decisions. However, these technologies are evolving and may not cover all the cases. For more detail of the methods employed, see SSE data Section 4.1.2 Han script.
4. There are some technical limitations found while integrating input from all experts which result in omitting some of the identified similar pairs. Such pairs are noted in SSE data Section 4.1.3 Neo Brahmi script and Section 4.1.4 Thai script.
5. In cases the SSE tool cannot process a certain string due to limitations in SSE data or SSE tool, a manual review of that applied-for string is needed.

5. Roles of the SSE Panel

The SSE panel will cover expertise in DNS, Internationalized Domain Names (IDNs), RZ-LGR, linguistics, and Unicode. The SSE panel will have additional experts to cover the scripts related to the strings, as needed.

AGB Section 7.10.1 defines the scope of comparison to cover the primary string and its variant strings against the existing gTLDs, gTLDs already in process, existing ccTLDs, strings requested as IDN ccTLDs, other applied-for gTLD strings, Blocked Names, two-character ASCII strings and their variant strings.

AGB also allows the SSE panel to decide whether and what blocked variant labels to omit when conducting a comparison. Such decisions must be based on SSE guidelines and criteria that justify such an omission on the basis of a manifestly low level of confusability between the scripts of strings being compared.

The SSE guidelines suggest the criteria for use by the SSE panel to decide on the omission of blocked variant labels when conducting a comparison.

The role of the SSE panel includes the following:

1. Follow SSE guidelines to do the complete evaluation of applied-for strings using the SSE data and the SSE tool.
2. Develop a uniform process for each panelist to prevent inter-panelist inconsistency and to ensure completeness and accuracy of the analysis.
3. Decide on the omission of any blocked variant labels when conducting a comparison based on the SSE guidelines.
4. Conduct the String Similarity Evaluation of applied-for strings based on the output report by the SSE tool, also considering the limitations of SSE tool and SSE data as noted.
5. Each applied-for string must be evaluated along with its variant strings. If a string does not appear in the SSE tool output report, its review must be done manually and the review must include the applied-for string as well as its variant strings.
6. Make the final decision on the String Similarity Evaluation for each applied-for string and its variant strings with one of the following possible outcomes:
    1. String Visually Similar to an existing gTLD or its variant string.
    2. String Visually Similar to an applied-for string in previous rounds of the New gTLD Program and still in process or its variant string.
    3. String Visually Similar to an existing ccTLD or its variant string.
    4. String Visually Similar to a requested IDN ccTLD or its variant string.
    5. String same as another applied-for string or its variant string.
    6. String Visually Similar to another applied-for string or its variant string.
    7. String Visually Similar to a Blocked Name or its variant string.
    8. String Visually Similar to a two-character ASCII string or its variant string.
    9. String not Visually Similar to any of these categories listed.

The similarity evaluation should include input from panelists who have native ability to read the scripts of the strings being evaluated.

The SSE panel should provide a rationale for their decision for contention sets which should include at least the following details:

1. Reference the relevant similarity category and relevant guidelines applied.
2. Explanation of any deviation from SSE tool results.

In cases where the SSE panel finds it is difficult to decide, the conservative approach should be considered and such cases should be marked as similar.

To assist the SSE panel, the guidelines on how to interpret the report from the SSE tool are provided in the section below.

6. Interpreting SSE Tool Output Report

The SSE tool generates a report which lists the potential similarity sets. The workflow of the SSE tool is available in Appendix A: Workflow of the SSE Tool.

Each set can contain two or more strings which are calculated to be similar based on the SSE data.

The pair of strings may be similar based on the applied-for string itself or its variant strings, even if those variant strings are not being applied-for. The pair of primary or variant string comparisons which cause the closest similarity will be listed and the similarity category of code points across the two strings is processed and documented. The SSE tool automatically excludes the comparisons between blocked variant strings, as these are out of scope of comparison based on the policy recommendation, as also indicated in the AGB Section 7.10.

These potential contention sets will also include information on direct contention (string A is confusable with string B) or indirect contention through string visual similarity transitivity (string A is confusable with string B and string B is confusable with string C but string A and string C are not confusable) or string-variant Visual Similarity transitivity (for example, string A is confusable with string B-variant-1 and string B-variant-2 is confusable with string C but string A and string C are not confusable).

The simplified version of the report is as shown in Figure 1 with mocked-up data for explanation purposes. The full report contains more details e.g. A-label and code points for each string, their variant disposition, etc.

Set	Applied-for String 1	Comparison String 2	Comparison String	Applied-for String 1 variant with most	Comparison String 2 variant with most	Similarity Category

			Source	similarity	similarity	
1	ɓát	bat	Another Applied-for gTLD	ɓát (primary label)	bát (blocked variant of bat)	2-1-1
2	ลาว	ລາວ	ccTLD	ลาว (primary label)	ລາວ (primary label)	2-2-2

Figure 1: Simplified SSE Tool Report

The SSE tool will also generate the Similarity Score which is calculated based on the Similarity Category values for each comparison. This score will help the SSE panel to sort or filter the report.

Among other factors, the SSE panel should carefully review the Similarity Category data in the last column in Figure 1, as produced by the SSE tool using the SSE data. The following sections include guidelines on how to interpret the SSE tool output report, based on the similarity categories listed in the 'Similarity Category' column.

6.1. The case of visually similar strings

The cases with high likelihood of visual similarity include the following (in all lowercase or all uppercase, as applicable):

1. A string is identical or near identical (all similarity categories are level 1) to another string.
2. A variant of the string is identical or nearly identical (all similarity categories are level 1) to another string.
3. A variant of the string is identical or nearly identical (all similarity categories are level 1) to a variant of another string, where the dispositions of both variant strings are not blocked.
4. A string is highly confusable (all similarity categories are level 1 or 2) with another string.
5. A variant of the string is highly confusable (all similarity categories are level 1 or 2) with another string.
6. A variant of the string is highly confusable (all similarity categories are level 1 or 2) with a variant of another string, where the dispositions of both variant strings are not blocked.

If the SSE panel overrides such a case to consider it not visually similar, the panel needs to provide a clear and strong rationale.

6.2. The case of probably visually similar strings

The cases with a likelihood of visual similarity include the following:

1. A string is similar (some similarity categories are level 1 or 2, with all remaining level 3) to another string.
2. A variant of the string is similar (some similarity categories are level 1 or 2, with all remaining level 3) to another string.
3. A variant of the string is similar (some similarity categories are level 1 or 2, with all remaining level 3) to a variant of another string, where the dispositions of both variant strings are not blocked.

If the SSE panel overrides any of the above cases to consider it not visually similar, the panel needs to have a clear and strong rationale.

4. A string is probably similar (all similarity categories are level 3) to another string.
5. A variant of the string is probably similar (all similarity categories are level 3) to another string.
6. A variant of the string is probably similar (all similarity categories are level 3) to a variant of another string, where the dispositions of both variant strings are not blocked.

If the SSE panel overrides such a case to consider it not visually similar, the panel needs to have a clear rationale.

6.3. The case of weakly visually similar strings

The cases which are not likely visually similar include the following:

1. A string is weakly similar (all similarity categories are level 1 or 2 or 3, but at least one level 4) to another string.
2. A variant of the string is weakly similar (all similarity categories are level 1 or 2 or 3, but at least one level 4) to another string.
3. A variant of the string is weakly similar (all similarity categories are level 1 or 2 or 3, but at least one level 4) to a variant of another string, where the dispositions of both variant strings are not blocked.

If the SSE panel overrides any of the above cases to consider it visually similar, the panel needs to have a clear rationale.

4. A string is not similar (all similarity categories are level 4) to another string.
5. A variant of the string is not similar (all similarity categories are level 4) to another string.
6. A variant of the string is not similar (all similarity categories are level 4) to a variant of another string, where the dispositions of both variant strings are not blocked.

If the SSE panel overrides such a case to consider it visually similar, the panel needs to have a clear and strong rationale.

7. Maintenance of the SSE Data and SSE Guidelines

Input on SSE data and guidelines may be sent to IDNProgram@icann.org.

Any additional input received at IDNProgram@icann.org after the publication of the SSE data will be recorded and discussed with the relevant experts. However, no further update will be made to the SSE data for the New gTLD Program: 2026 Round.

ICANN will compile input received at IDNProgram@icann.org after the publication of the SSE guidelines, as well as input received from the SSE panel. However, no further update will be made to the SSE guidelines for the New gTLD Program: 2026 Round.

The String Similarity Evaluation Data (version 1.0, dated 23 July 2026) and this String Similarity Evaluation Guidelines for the New gTLD Program: 2026 Round (version 1.0, dated 23 July 2026) are the applicable versions for the New gTLD Program: 2026 Round.

8. Conclusion

The document provides SSE guidelines for the SSE panel based on the related policies and AGB, and is intended to be used by the SSE panel along with the SSE tool and SSE data to make the final String Similarity Evaluation decisions.

These guidelines provide the details of the scope of similarity evaluation, functions of the SSE tool and SSE data, and how to interpret the report generated from the SSE tool. However, it remains the responsibility of the SSE panel to make the final decision and determination on the String Similarity Evaluation for all applied-for strings. The SSE panel needs to provide rationale for its decisions.

The string similarity evaluation by the SSE panel will be done during the String Evaluation stage of application for a gTLD string during the New gTLD Program: 2026 Round.

9. References

- Public Comment proceeding of the String Similarity Evaluation Guidelines for New gTLD Program: 2026 Round, 26 November 2025 https://www.icann.org/en/public-comment/proceeding/string-similarity-evaluation-guidelines-for-new-gtld-program-2026-round-26-11-2025.

- Public Comment proceeding of the String Similarity Evaluation Data, 10 October 2025, https://www.icann.org/en/public-comment/proceeding/string-similarity-evaluation-data-for-new-gtld-program-next-round-16-10-2025.
- Root Zone Label Generation Rules Version 6 (RZ-LGR-6), 23 September 2025, https://www.icann.org/resources/pages/root-zone-lgr-2015-06-21-en.
- ICANN org, String Similarity Review Guidelines, 19 March 2024, https://itp.cdn.icann.org/en/files/strategic-initiatives/public-comment-summary-report-string-similarity-review-guidelines-15-05-2024-en.pdf.
- GNSO Phase 1 Final Report on the Internationalized Domain Names Expedited Policy Development Process, 8 November 2023, https://gnso.icann.org/sites/default/files/policy/2023/correspondence/epdp-idns2-leadership-team-et-al-to-gnso-council-et-al-08nov23-en.pdf.
- GNSO Final Report on the new gTLD Subsequent Procedures Policy Development Process (Topic 24), 2 February 2021, https://gnso.icann.org/sites/default/files/file/field-file-attach/final-report-newgtld-subsequent-procedures-pdp-02feb21-en.pdf.

Appendix A: Workflow of the SSE Tool

The SSE tool generates a pre-screening report of potential contention sets of applied-for strings with the following workflow.

Input:

1. New gTLD Program: 2026 Round applied-for strings
2. List of strings in relevant categories as per Applicant Guidebook, Section 7.10, i.e,
    a. Existing gTLD
    b. Applied-for string from previous round(s) of the New gTLD Program still in process
    c. Existing ccTLD
    d. Requested IDN ccTLD
    e. Blocked Name
    f. Any two-character ASCII
3. SSE data
4. RZ-LGR

Process:

1. Identify the potential contention set by using SSE data to calculate the SSE index for all strings in items 1 and 2 and their variant strings in the Input section above. The SSE Index is a unique value which represents all the strings in a string similarity set.

2. Group the strings which have the same SSE index in the same potential contention set. There may be two or more strings in a set.
    Example: Set 1 includes *string-A* and *string-B*.
    Set 2 includes *string-X*, *string-Y*, and *string-Z*.

3. For each set, pair up all strings in the set.
    Example: The pairs in Set 1 are:
    string-A and *string-B*
    The pairs in Set 2 are:
    string-X and *string-Y*,
    string-X and *string-Z*,
    string-Y and *string-Z*.

4. For each pair, generate variant strings using the RZ-LGR-6 and compare all members of the two strings' variant members.
    Example: *string-A* has one variant *string-Av1*,
    string-B has two variants *string-Bv1* and *string-Bv2*
    (assuming all variant strings are allocatable)

    The six comparisons are performed and provide the similarity category mapping based on the category in the SSE data (see Section 3 Category and Similarity Level). The mock-up result of category mapping for each comparison is listed below.

string-A and *string-B* have the similarity category mapping = 1-2-4
string-A and *string-Bv1* have the similarity category mapping = 1-2-4
string-A and *string-Bv2* have the similarity category mapping = 1-2-3
string-Av1 and *string-B* have the similarity category mapping = 1-2-3
string-Av1 and *string-Bv1* have the similarity category mapping = **1-2-1**
string-Av1 and *string-Bv2* have the similarity category mapping = 1-2-3

The lower category means the higher level of visual similarity. Therefore, the pair *string-Av1* and s*tring-Bv1* are the most visually similar.

(Note: any blocked variant string with blocked variant string comparisons will be omitted by the tool)

5. Include the most similar case(s) from step 4 in the report (i.e. *string-Av1* and *string-Bv1* (1-2-1) in the example above). The Admin view displays only one most similar case while the Panelist view displays all cases, ordered by the level of similarity (see also limitations on the analysis based on the number of variant strings in Section 4).

Output:
The pre-screening report includes the list of all potential contention sets. Each set provides the comparison pairs between strings within the set along with the details of which primary or variant strings cause the most similarity level. The report is generated in both HTML format and CSV format for the SSE panel to conduct further review by the SSE panel.

Based on the above example, the following line will be added to the report (Admin View) for Set 1, as shown in Figure A.1. It is interpreted as "*string-A and string-B are in potential contention and string-Av1 and string-Bv1 are most similar (1-2-1).*"

Set	Applied-for String 1	Comparison String 2	Comparison String Source	Applied-for String 1 variant with most similarity	Comparison String 2 variant with most similarity	Similarity Category
1	string-A	string-B	Another Applied-for gTLD	string-Av1	string-Bv1	1-2-1

Figure A.1: Simplified SSE Tool Report for string-A and string-B Example (Admin View)

The following lines will be added to the report (Panelist View) for Set 1, as shown in Figure A.2. It is interpreted as string-A and string-B are potential in contention and all variant string set members comparisons are displayed.

Set	Applied-for String 1	Comparison String 2	Comparison String Source	Applied-for String 1 variant with most similarity	Comparison String 2 variant with most similarity	Similarity Category
1	string-A	string-B	Another Applied-for gTLD	string-Av1	string-Bv1	1-2-1
				string-A	string-Bv2	1-2-3
				string-Av1	string-B	1-2-3
				string-Av1	string-Bv2	1-2-3
				string-A	string-B	1-2-4
				string-A	string-Bv1	1-2-4

Figure A.2: Simplified SSE Tool Report for string-A and string-B Example (Panelist View)

Both the Admin View and Panelist View of the report will be available for the SSE panel.