



Proposed Solution for Writing Domain Names in Different Arabic Script Based Languages

ASIWG 2nd Meeting, Dubai May 28-29 2008

Dr. Abdulaziz H. Al-Zoman
Director of SaudiNIC - CITC
Chairman of Steering Committee
Arabic Domain Name Pilot Project

Raed I. Al-Fayez
SaudiNIC - CITC
Chairman of Technical Committee
Arabic Domain Name Pilot Project



Agenda

- About Arabic script
- Major Issues
- Registry Challenges!
- Characteristics of A Desired Solution
- Basic Assumptions
- What Each Ready Language Should have
- Combing the languages work
- Steps needed by the Registry
- To support a new language



About Arabic script



- The **2nd** most widely used alphabetic writing system in the world (used by more than **43 countries**)
 - more than **one billion potential users** could be concerned in using Arabic script domain names.
- Used by **many languages** such as: Persian, Urdu, Turkish, Kurdish, Pashto, Swahili, ...
 - that may **add or change** characters to represent phonemes that do not appear in Arabic phonology.
 - A new character usually created based on **modifying the basic shape** of an existing Arabic character, for example, by adding more dots.
 - These additions have **meaning to the new language** but not to the original language.
 - Therefore, many characters would **easily be confused** with some other characters from other languages



Unicode Arabic Block (5.1)



- | | |
|--|---------------------------------------|
| ▪ Subtending marks | ▪ Other combining marks |
| ▪ Radix symbols | ▪ Arabic-Indic digits |
| ▪ Letterlike symbols | ▪ Archaic letters |
| ▪ Punctuation | ▪ Points |
| ▪ Currency Signs | ▪ Extended Arabic letters |
| ▪ Poetic marks | ▪ Extended Arabic letters for Parkari |
| ▪ Honorifics | ▪ Eastern Arabic-Indic digits |
| ▪ Koranic annotation signs | ▪ Signs for Sindhi |
| ▪ Based on ISO 8859-6 | ▪ Additions for Khowar |
| ▪ Addition for early Persian and Azerbaijani | ▪ Additions for Torwali |
| ▪ Points from ISO 8859-6 | ▪ Additions for Burushaski |
| ▪ Combining Madah and Hamza | ▪ Additions for early Persian |

298 Code points



Arabic Script IDN - Major Issues

- **Acceptable/disallowed characters**
 - IDNA200x table (Pvalid / Disallowed / ContextO)
 - Language tables
- **Non-spacing Marks**
 - Subtending Marks (U+0600 - U+0603)
 - Honorifics (U+0610 - U+0614)
 - Koranic annotation signs (U+0615 - U+061A, U+06D6 - U+06ED)
 - Points (U+064B - U+0652, U+0670)
 - Combining Maddaa and Hamza (U+0653 - U+0655)
 - Other combining Marks (U+0656 - U+065E)
- **Confusing similar characters (e.g. variant tables)**
- **World/label separators (space, ZWNJ, ZWJ, hyphen)**
- **Bidirectional**
- **They are addressed at different levels:**
 - IDNA protocol level
 - Registry level
 - Application level



Arabic Script IDN - Registry Challenges!

- **Security issues** (stability, trust,...) e.g. phishing
 - They should be addresses at language level first
- **Not all Arabic-based languages are ready:**
 - Not widely/commonly used
 - Language community are not ready
- **Hard to make decisions** on behave of other language communities
- **Pressure** to start with ready languages
- Many **problems** have been **escalated** from the protocol to be handled by the registry (e.g. variants, bundling ..etc)
- ... and yet has to provide a **simple** and **transparent** registration services



Registration Form (demo) نموذج التسجيل (تجريبي)

Invalid option (position) Mixed of languages Can be enabled

Choose a Language عربي اختر اللغة
 Type a domain name هيئة اسم النطاق
 TLD .مثال انطاق العلوي

Clear مسح Search ابحث

Number of possible matches including requested domain: 64

Difficult to manage

(U+0629)ة	(U+0626)ى	(U+064A)ي	(U+0647)ه	هيئة.مثال
(U+0629)ة	(U+0626)ى	(U+064A)ي	(U+06BE)هـ	هيئة.مثال
(U+0629)ة	(U+0626)ى	(U+064A)ي	(U+06C1)هـ	بيئة.مثال
(U+0629)ة	(U+0626)ى	(U+064A)ي	(U+06D5)هـ	ديئة.مثال
(U+0629)ة	(U+0626)ى	(U+067B)ب	(U+0647)ه	هبيئة.مثال

Done

Arabic Script IDN -Who is ready?

- Some language communities are **somehow ready** (alphabetical order):
 - Arabic
 - Jawi
 - Pashto
 - Persian
 - Urdu
 - ...

Characteristics of A Desired Solution



- Based on **standardized** (or agreeable) policies and procedures that are documented on RFC-like or Best-Practice documents
- **Extendable** to allow for adding new languages as they become ready
- **Simple and transparent** to the end user (**registrant** and **navigator**)
- **Easy and fast** to be deployed by any registry
- Work for both **ccTLDs** and **gTLDs**



Basic Assumptions



- **Agreeable** solutions for the major issues have been reached ... that will
 - adhere to **LDH** convention
 - follow the **inclusion** mode recommended by the new IDNA standards
 - flexible to **cover changes/updates** to the Arabic block in Unicode
 - Properties of code points
 - Normalization



A Ready Language Should have




- **Language Table**
 - A simple code point set table that **includes ONLY needed** letters and digits
- **Variant tables**
 - **Exact Variant Table (EVT)**
 - A comprehensive variant table that includes all codepoints (from the whole script) that are **confusingly similar** (exact match)
 - Example: ف
 - **Typo Variant Table (TVT)**
 - A comprehensive variant table that includes all codepoints (from the whole script) that **may confuse** end users (e.g. **typo/style match**)
 - Example ی
 - **Note:** A variant table will consist of a **list of records**, each record contains the following information:
 - Basic character codepoint,
 - variant codepoint,
 - positions of similarity [Standalone, End, Middle, Beginning]





Examples of variants



▪ Example **Exact** Variant match

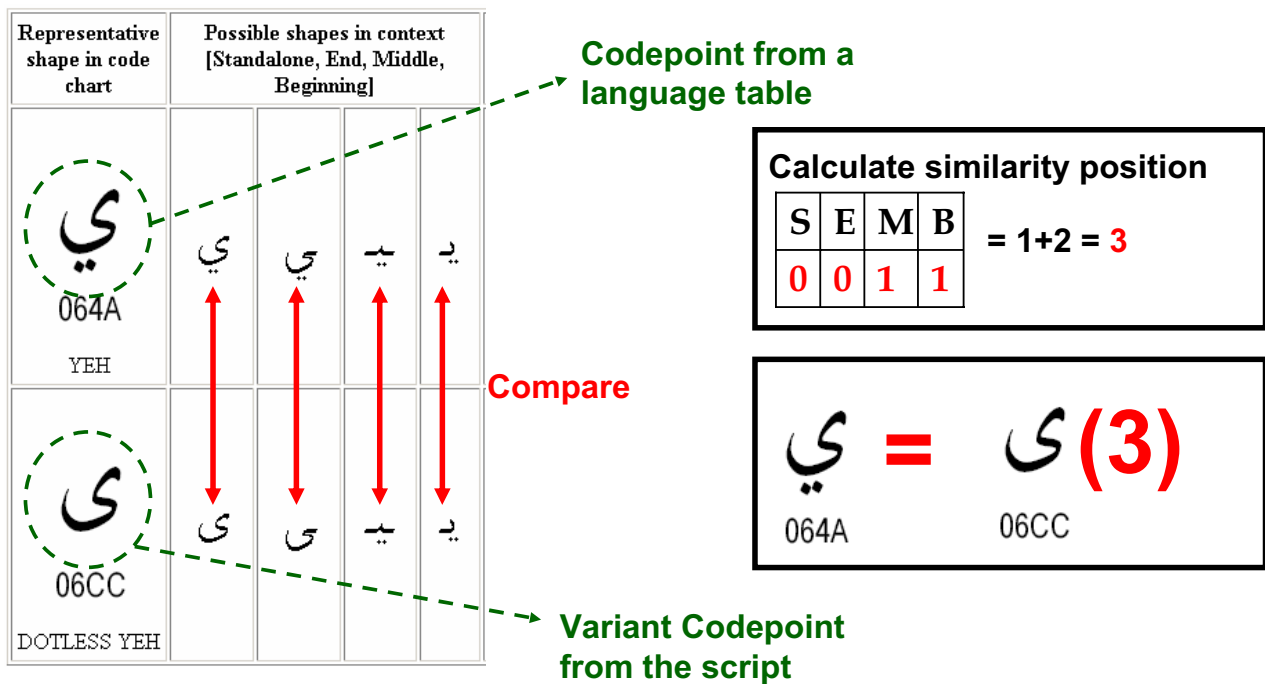
Representative shape in code chart	Possible shapes in context [Standalone, End, Middle, Beginning]			
 0641 FEH	ف	ف	ف	ف
 06A7 QAF WITH DOT ABOVE	ق	ق	ق	ق

▪ Example for **Typo** Variant match

Representative shape in code chart	Possible shapes in context [Standalone, End, Middle, Beginning]			
 0649 ALEF MAKSURA	ا	ا	ا	ا
 06CD YEY WITH TAIL	ي	ي	ي	ي



Example: Defining the position of similarity

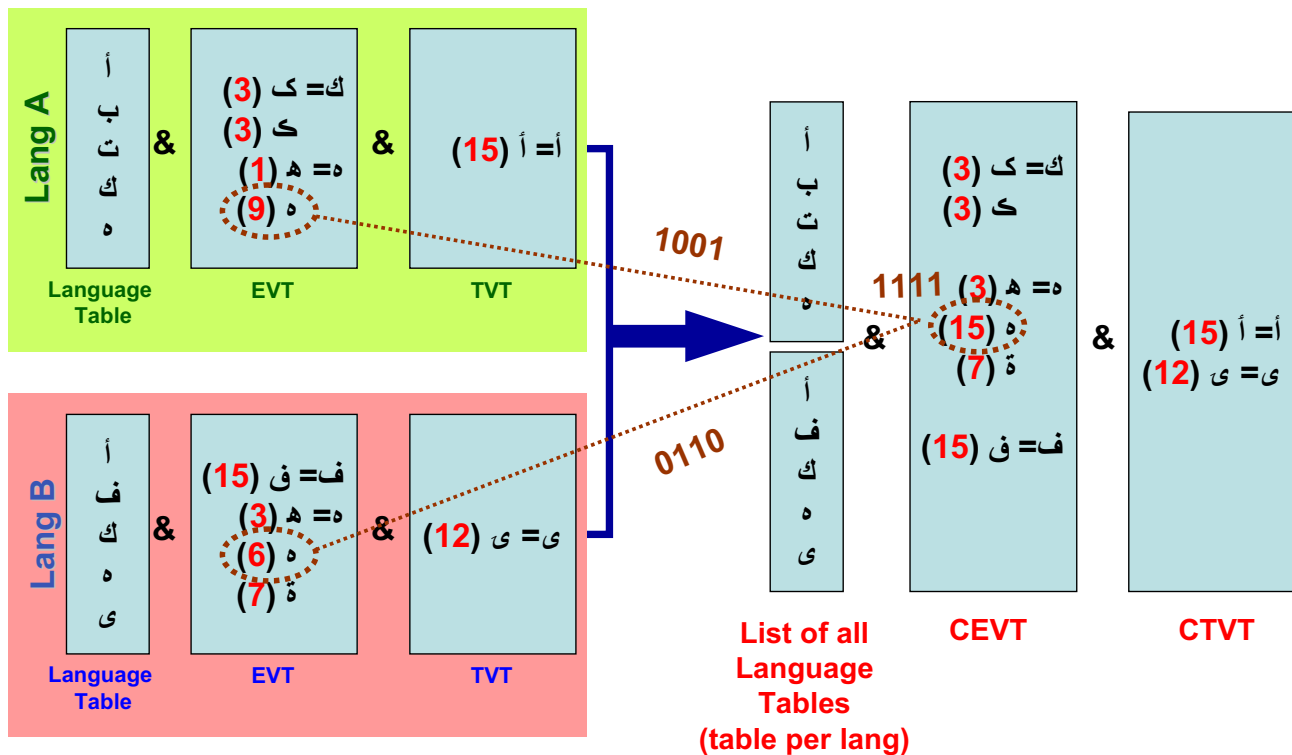


Build Registry Script-based Tables

- Have a “**Language Table**” for **each supported language**
- Build a “**Combined Exact Variant Table**” (CEVT) from exact variant tables (EVT) of the supported Languages
- Build a “**Combined Typo Variant Table**” (CTVT) from typo variant tables (TVT) of the supported Languages
- **Note:**
 - When combing variant tables use “**OR**” operation on position of similarity



Example: Combing the languages work



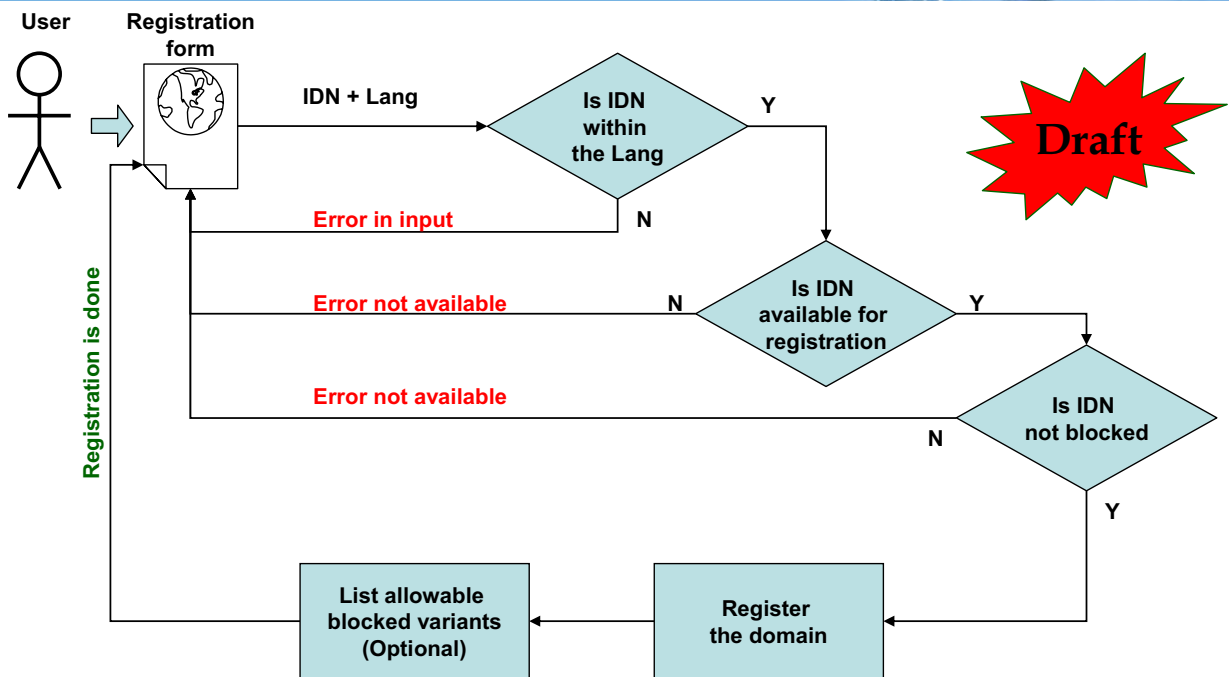
Registry-Registrant interface



- **Registration process:**
 - Registry lists supported languages
 - Registrant should select one of these languages and type a domain (U-Label)
 - Registry should accept inputs based on the selected language table
 - If domain name is register-able (available and not blocked)
 - Register the domain name
 - List of allowable blocked variants
- **Activation process (variants)**
 - Original registrant can activate allowable blocked variants
 - Allowable blocked variants are variants that are generated from supported languages without intermingling between languages tables
- **Lookup process (on-the-fly)**
 - Direct Whois
 - If domain is available or not
 - Full list of all blocked variants (e.g. reference-domain)
 - List all possible variants using CEVT & CTVT
 - Take care of confusion position of variant (optimization)
 - Only allowable blocked variants
 - Filter the full list to get a list of variants that are generated from supported languages without intermingling between languages tables



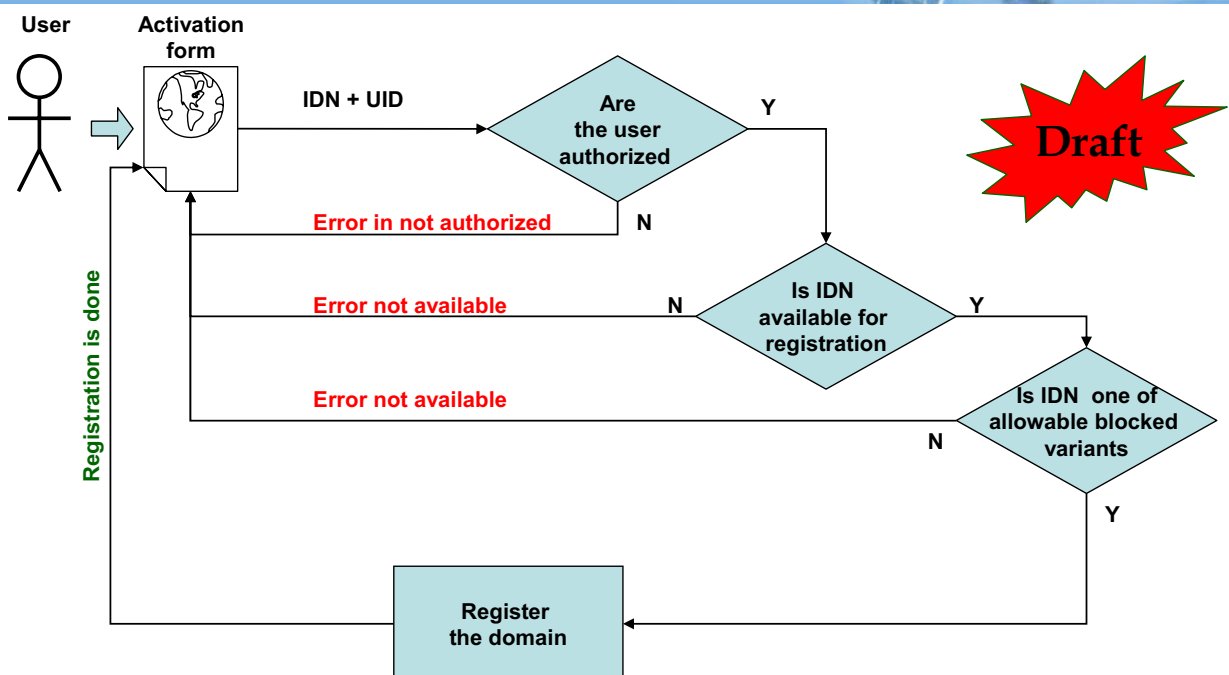
Steps needed by the Registry



Can be on the fly



Steps needed by the Registry



Can be on the fly



To support a new language

- **Need process owners !!!!**
- **Language community should prepare**
 - **Language Table**
 - **Exact Variant Table (EVT)**
 - **Typo Variant Table (TVT)**
- **Support the new language:**
 - Add it to the **list** of supported languages
 - Add their **Language table**
 - Regenerate **CEVT & CTVT**
- **Communicate it with others (ICANN, registries ..)**



Conclusion

- **These are initial thoughts**
 - If we agree on this model we can put more efforts to have a detailed structure/solution





Thank you

شكرا

شكرا

xn--mgbti28b

input[0] = U+0634
input[1] = U+06a9
input[2] = U+0631
input[3] = U+0627

xn--mgbti4d

input[0] = U+0634
input[1] = U+0643
input[2] = U+0631
input[3] = U+0627