Built to Last
Goals

1. Study, review, and document the current tamil unicode representations
2. identify the stability issues wrto TACE-16
3. identify solutions to bridge limitations of (1) with the advantages of TACE-16
4. Identify ways to accomodate TACE-16 in BMP
5. Identify ways to interoperate with TACE-16 (interoperable standard)
Indic Unicode

- Based on ISCII -1988
- Multiple code points to render single characters
- Requires ZWJ/ZWNJ type hidden chars
- Requires complex collation tables, normalization
- Sorting, Searching, counting, inefficient
- Needs exception table to prevent illegal combinations of code points
- Subject to revision if combinations are permitted
Built to Last
ISCII and Unicode

- ISCII was designed for 8 bit world in 1988
- ISCII’s purpose was limited – xlit across Indic
- ISCII did not anticipate that it will be encoding all Indic languages for all future technologies
- Unicode Indic block built an enormous, complex, error-prone edifice based on an encoding that was NOT built to last
Built to Last?

Based on ISClI 1988

Various signs

0B82  உ TAMIL SIGN ANUSVARA
  • not used in Tamil

0B83  ஊ TAMIL SIGN VISARGA
  = aytham

"Don't tell me. Let me guess."
Tamil Unicode — Built to last?

- Not designed by the language community
- Designers not familiar with Tamil
  - Very first code point says “Tamil Sign Anusvara — Not used in Tamil”
  - Next sign says “Tamil Sign Visarga” and makes it a dependent letter — corrected later
  - Assumed collation was same as Devanagari - incorrectly
- Uses ambiguous encoding to render same character
- Encodes Vowel-Consonant and calls it a consonant
Limitations of Indic/Tamil Unicode

- Requires multiple code points for the most used characters
- Codepoint bloat — doubles the size, costly
- Multiple code points lead to
  - Security vulnerabilities
  - Ambiguous combinations — requires normalization
  - Simple counting, sorting, searching inefficient
- Encodes display specific rules as characters
Advantages of 16-bit Indic/Tamil

- Single code point for single letter
- Display attributes should be left to the font
- May not need ZWJ/ZWNJ type hidden chars
- No built-in security vulnerabilities that need to be fixed with exceptions – no need to split chars across codepts
- Sorting, searching, counting etc., are straightforward
- Storage, bandwidth requirements reduced
- Offers stability
Korean Argument

- Unicode has given to Korean language, both the canonical version (equivalent to Tamil Unicode 5.0) AND a syllable version (equivalent to “New Tamil”)
  - The respective code plans even specify equivalence
  - Why?

- The “Half Jumo” and “Half Katakana” plane FF00-FFEF even has English Characters!
  - Why?
Hangul & Jamo

- **Hangul** [AC00-D7A3]
  - Syllable Block
  - L+V or L+V+T

- **Jamo** [1100-11FF]
  - Primitives
    - L: Leading Consonant
    - V: Vowel
    - T: Trailing Consonant

- **Hangul Characters can be composed with Jamo**
Let us build to Last
Where Tamil Nadu is going

- TACE-16 is on State and National standards track
- TACE-16 is efficient, stable, backwards compatible with characters across centuries
- TACE-16 addresses Tamil IT needs efficiently
- TACE-16 standard support will be required for Government purchase
- Tamil is the first of the Indic languages with 16-bit encoding
- Recognition of Indian national standards by International standards bodies is reasonable
Where we would like Unicode to go

- Acknowledge the fairness of a language community to set its own standard and recognize it
- We have spent a lot of time and money to analyze the data and found that Unicode Tamil is inefficient
- Give a fair hearing to the complaints that have been strong and loud for over 8 years
- Find ways to accommodate TACE-16 in the BMP
- At least include missing vowel-consonants and consonants as pre-composed syllabic characters in BMP
Thank You

Thank you