



HAL
open science

DHARMA Transliteration Guide

Dániel Balogh, Arlo Griffiths

► **To cite this version:**

| Dániel Balogh, Arlo Griffiths. DHARMA Transliteration Guide. 2020. halshs-02272407v3

HAL Id: halshs-02272407

<https://hal.science/halshs-02272407v3>

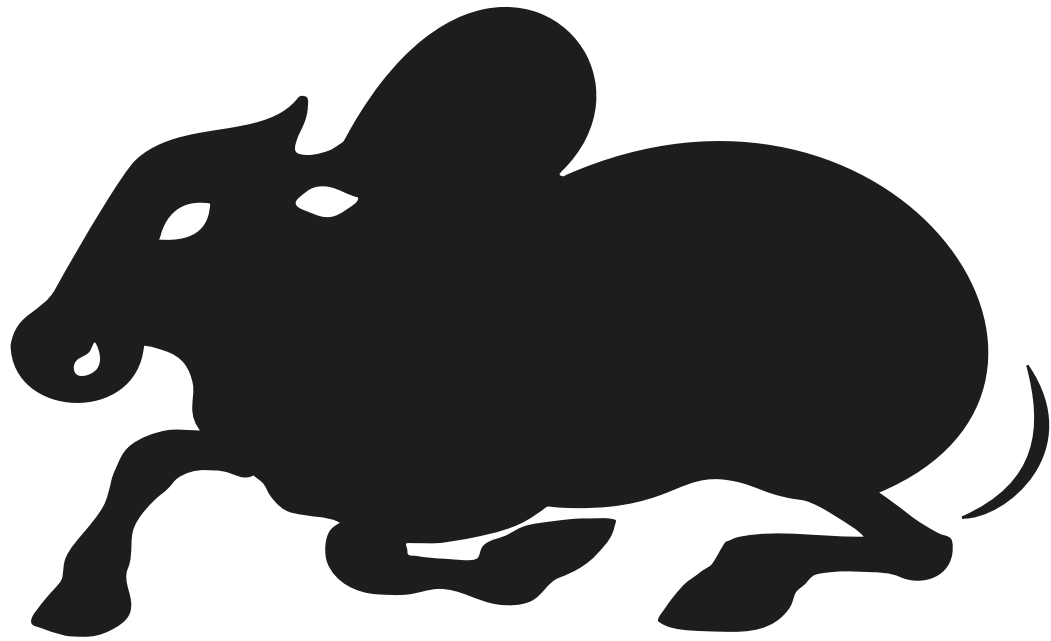
Preprint submitted on 2 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License



dharmā

Transliteration Guide

Dániel Balogh & Arlo Griffiths

Release Version 3, 2020-07-05



This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 809994).

Contents

1. Introduction	2
1.1. Version History	2
1.1.1. Summary of changes since the last version	2
1.2. Coverage	2
1.3. Separation of Transliteration and Encoding	2
1.4. Terms and Definitions	3
1.4.1. Abbreviations	3
1.4.2. Script and its elements	3
1.4.3. Script conversion	5
1.4.4. Notation for transliteration and transcription	6
2. General Principles	7
2.1. Character Set and Input Method	7
2.2. Transliteration in Practice	8
2.2.1. Strict transliteration	8
2.2.2. Loose transliteration	8
2.2.3. Shorthand	9
2.3. Transliteration Scheme	9
2.4. Case Sensitivity	9
2.4.1. A note on the use of uppercase for standalone vowels and consonants	10
2.5. Disambiguation	10
2.6. Editorial Additions for Text Analysis	11
2.6.1. Editorial spaces for word segmentation	11
2.6.2. Editorial hyphenation	12
2.6.3. Representation of <i>avagraha</i>	13
2.6.4. Representation of elided overshoot final <i>u</i> in Tamil	14
3. Alphabetic Characters	15
3.1. Some Special Characters	15
3.2. Long and Short e and o	16
3.3. Special Glyph Forms and Compositions	16
3.3.1. Final consonants as special simplex characters	16
3.3.2. Final consonants as complex characters involving a zero vowel marker	17
3.3.3. Independent vowels as special simplex characters	17
3.3.4. Independent vowels as complex characters involving a “vowel support”	17
3.3.5. Multiple vowel markers within an <i>akṣara</i>	19
3.3.6. Repurposed vowel markers	19
3.3.7. Short vowel written where a corresponding long vowel is expected	20
3.3.8. Unusually composed complex characters	20
3.3.9. Characters with alternative or optional phonemic values	21
3.3.10. Complex characters split by an intervening feature	21
3.3.11. Special forms of <i>anusvāra</i>	22
4. Non-alphabetic Characters	23
4.1. Numerals	23
4.1.1. Numbers denoted by bars	23
4.1.2. Fractions	24
4.2. Symbols	24
4.2.1. Punctuation marks	24
4.2.2. Space filler signs	25
4.2.3. Generic symbols	26
4.3. Space	26
References	27

1. Introduction

1.1. Version History

Author(s)	Version	Changes	Date
Balogh, Griffiths	0.1	First draft	2019-07
Balogh, Griffiths	1	Expansion and revision for first release	2019-09
Balogh	2 (1.1) ¹	Revision	2019-12
Balogh, Griffiths	3	Revision and expansion	2020-07-05

1.1.1. Summary of changes since the last version

- deprecated ° as a marker for independent vowels and final consonants (§2.4.1)
- added §3.3.10 on *Complex characters split by an intervening feature*
- restricted the use of : to ISO-15919 disambiguation (§2.5) and the Indonesian *tarung/tedong* (§3.3.6)
- introduced the use of = (to replace the former :) for flagging unusual *akṣara* composition (§3.3.8) including Tamil ligatures and varying reading modes of superscript *r*
- extended the use of + in numeral notation to cover fractions (§4.1.2)
- revised the transliteration of vowel supports (§3.3.2)
- revised transliteration of special *anusvāra* glyphs (§3.3.11), including the Javanese/Balinese form with an additional stroke
- substantially revised method of handling symbols (§4.2)

1.2. Coverage

This Guide is essentially intended to cover the scripts relevant to the languages with which the DHARMA project is concerned, i.e., in alphabetical order (omitting the adjective “Old” relevant in several cases): Balinese, Cam, Javanese, Kannada, Khmer, Malay, Prakrit, Sanskrit, Sundanese, Tamil, Telugu. However, the recommendations we give here are certainly intended to be compatible with and extensible to other languages and scripts. We request from colleagues reading and using this text to draw our attention to phenomena in the covered languages/scripts that we have so far failed to address, and to give suggestions on how they might be integrated, as well as to phenomena in languages/scripts so far not covered that may cause issues of compatibility.

The contents of this Guide are primarily applicable to digital editions of epigraphic texts, which must follow these instructions rigorously. We do however hope (and, to some degree, expect) that project members will use the same transliteration method, as far as applicable, in their print publications and other work. Section 2.2 gives some further pointers on what features of the transliteration system can be ignored outside diplomatic editions.

1.3. Separation of Transliteration and Encoding

When digitally representing the text of inscriptions (and manuscripts) for preservation and for computer-aided research, we strive to keep recorded content (i.e. what text is written on a certain support) separate, or at least separable, from our annotations *describing* various aspects of that content (for instance how it is written and laid out, how clearly it is readable, or what sort of information it carries). Content is transliterated according to the methods covered in this Guide, while descriptive annotation is added in the form of EpiDoc markup as detailed in the Encoding Guide.² The same descriptive annotation also plays a

¹ Version 2 bears the internal version number 1.1, but since the automatic versioning in the HAL-SHS repository assigned it number 2, we have chosen to adopt that numbering to eliminate future inconsistencies. In the future, release versions deposited on HAL-SHS will have integer numbers, but project-internal versions (stored on GitHub) may have fractional ones.

² We follow the TEI Guidelines in using the terms ‘markup’ and ‘encoding’ as interchangeable synonyms.

role in determining how our text will be ultimately presented to users on screen and in print, but this is yet another separate concern and will not be addressed here.

Ideally, therefore, no issues that pertain to the description of the physical manifestation of a text should be recorded in the transliterated text itself; and likewise, no issues that pertain to the text content should be omitted from the transliterated text and recorded only in markup. In practice, there are a number of borderline cases that could arguably belong to either of these domains. Given that we are primarily concerned with the faithful documentation of epigraphic texts, some of these issues (such as the use of dedicated signs for independent vowels and final consonants) are addressed at the level of transliteration, while others (such as the possibility of interpreting an ambiguous glyph as either of two or more characters) are dealt with in markup. There is inevitably a certain degree of fuzziness and permeability at the boundary between these domains. Some of the phenomena we cover in transliteration (because we feel that this makes the encoders' job easier) will be universally and automatically converted to markup, and some others may at a later time be likewise converted.

It should be apparent from this that transliteration and markup go hand in hand. We hope that everyone involved in digitising texts will acquire a working familiarity with both Guides, and that even those who will not be creating fully marked-up EpiDoc editions will be willing and able to add snippets of markup to their texts to cover phenomena that cannot be handled through transliteration alone. Cross-references between the Guides should help you find the correct way to deal with each case.

1.4. Terms and Definitions

1.4.1. Abbreviations

In addition to common abbreviations, this Guide uses:

- TG the DHARMA Transliteration Guide (the present document)
- EGD the DHARMA Encoding Guide for Diplomatic Editions (version 1.0)³

1.4.2. Script and its elements

- a **script** may be defined as “a set of conventional graphic signs designed to give visual representation to the elements of a writing system” (Wellisch 1978, 15)
 - here, a **graphic sign** is defined as “any conventional mark by which a human being intends to affect the state or behavior of other human beings” (ibid. 10)
 - and a **writing system** is defined as “a system of rules governing the recording of words and sentences of a language by means of conventional graphic signs” (ibid. 13)
- in the usage of this Guide,
 - **Latin script** refers to the family of fully alphabetic scripts used for writing most European and many other languages
 - the term Roman script is sometimes used in an equivalent sense, but we prefer to designate it as Latin here because Unicode and ISO do so, and because Roman is used in typography to designate a specific set of typefaces within the Latin script
 - **Indic script** refers to the family of alpha-syllabic scripts derived from the Brāhmī script and used for writing most historic South and Southeast Asian languages
- the term **character** may be defined in several ways
 - according to Wellisch (1978, 16), “A **character** is an element of a script, representing a phoneme, syllable, word, or prosodic feature of a language by means of graphic signs.”
 - for our purposes we prefer to emphasise, with Ollett and Taylor (forthcoming), that a character is “an element of the writing system that can be used independently according to the logic of that writing system”
 - thus, Latin letters such as *a*, *b*, *c* are each one character, and one character represents no more than one phoneme

³ References to the EGD in this document pertain to EGD version 1, released simultaneously with TG version 3. The EGD release will be published on HAL-SHS, and the working internal document (with ongoing revisions) will be available in the

- some phonemes are represented in some writing systems by a combination of several characters, e.g.
 - English *th* (representing either the voiced dental fricative /ð/ as in ‘this,’ or the voiceless dental fricative /θ/ as in ‘thing’)
 - ISO15919-transliterated Indic *th* (representing in Sanskrit the aspirated voiceless dental plosive /tʰ/ as in *ratha*)
- such combinations are technically called **polygraphs** or, when exactly two characters are involved, **digraphs**
- however, in an Indic writing system, one *akṣara* is one character
 - regardless of how many phonemes it represents and how many visually and semantically distinguishable parts it consists of
 - e.g. Devanagari उ, क्, क, कि and क्ख are each one character
 - while none of the elements corresponding to the transliterated characters *r*, *d*, *dh* and *e* in the *akṣara* क्ख are themselves characters (we refer to these as components, see below)
 - to reduce ambiguity, characters such as उ and क् may be called **simplex characters**, while characters such as कि and क्ख may be called **complex characters** (and note that characters such as क् could arguably belong to either of these classes)
 - strictly speaking, *anusvāra* and *visarga* are not characters by this definition
 - however, we do not foresee a need to classify them rigorously, and believe that in some circumstances it may be more productive and intuitively correct to think of these signs (especially *visarga*) as characters
- some characters (in any writing system) have a semantic value that does not correspond directly to any phonemes, e.g.
 - numeral signs are definitely characters
 - punctuation signs and other symbols used in written text are arguably characters, and we prefer to include them in the scope of the term
 - to reduce ambiguity, the terms **alphabetic character** and **non-alphabetic character** may be used to distinguish between these subsets
- a character defined as above is essentially equivalent to a **grapheme**, often defined as “the smallest functional unit of writing on whatever structural level of language the writing system operates” (Coulmas 2006, s.v.)⁴
- in information and computer science, a **Unicode character** is an abstract element of the script, defined as a “member of a set of elements used for the organization, control, or representation of textual data” (ISO/IEC 10646:2017(E), 2)
 - this technical definition is not something we need to use regularly, but it is good to be aware that this definition of a character includes:
 - entities with a visual counterpart (graphic characters) that *represent* phonemes or other information (e.g. punctuation)
 - thus, in this sense of character, the *akṣara* कि = क् = क्ख *ki* consists of two characters, the abstract *k* and the abstract *i*
 - as well as functional characters that do not necessarily have a visual counterpart and exercise *organization* and *control* over graphic characters; for instance in Indic scripts
 - conjunct consonants such as Devanagari क्क involve a non-graphic *virāma* character whose function is to tell the computer that the graphic characters are to form a conjunct (ligature)
 - unusually formed conjuncts such as Devanagari क्क include, in addition, a control character called a zero-width non-joiner to tell the computer that this particular *virāma* should not form a conjunct (the expected Devanagari क्क), but manifest as a visible zero vowel marker
 - a **glyph** is a concrete graphical representation of any particular character
 - thus the Indic character *ma* may be represented by the glyphs ए, म, ङ, ञ, ङ etc.

⁴ The term ‘grapheme’ is sometimes defined differently, so that polygraphs are considered to be a single grapheme; this definition does not concern us here.

- Unicode parlance prefers to use the term **graphic symbol**, defined as the “visual representation of a graphic character or of a composite sequence” (ISO/IEC 10646:2017(E), 5)
- another roughly synonymous term is **graphic sign**, defined as “any conventional mark by which a human being intends to affect the state or behavior of other human beings” (Wellisch 1978, 10)
- yet another quasi-synonym is **graph**, defined as “The smallest formal unit of written language on the level of handwriting or print” (Coulmas 2006, s.v.)
- visually different glyphs representing the same character within a writing system are known as **allographs**
 - e.g. in the Latin script, the glyphs ‘a’ and ‘ɑ’ are allographs (and, for most practical purposes in most languages, a and A are likewise allographs)
- to refer to parts of complex Indic characters that are visually distinct and have a semantic value of their own, we use (and encourage the use of) the term **component**; thus,
 - **character components** are elements such as those representing the phonemes *r*, *d*, *dh* and *e* in the Indic character *rddhe*, as well as the zero vowel marker in the Indic character *k* composed with an explicit vowel killer
 - while **glyph components** are particular realisations of character components in any specific script, such as the stroke combinations corresponding to the transliterated characters *r*, *d*, *dh* and *e* in Devanagari ढ्ढ्, or those representing *ka* and the zero vowel marker in Devanagari क्
 - when no distinction between character and glyph is required, “component” may be used on its own to refer to these entities
 - components which can never occur independently, but which can occur in combination with various other components, may be specifically called **markers** (with Ollett and Taylor forthcoming)
 - in Indic scripts these include in particular dependent vowel markers and zero vowel markers, but some other signs, such as the *upadhmāṇiya* and *jihvāmūliya*, the *repha*, and arguably also the *anusvāra* and *visarga*, may also be included in the scope of this term
 - note that the term “component” is sometimes (e.g. Brookes et al. 2015, 34) also used to refer to distinctive subunits of non-complex characters, i.e. to elements without phonemic correspondence
 - although it is not relevant to this guide, we recommend avoiding the word “component” in this sense and instead encourage the use of **stroke** to refer e.g. in palaeographic descriptions to the visual elements that make up a character and to their graphic manifestations that make up a particular glyph
 - we also encourage the use of biological and architectural analogues to describe particular strokes, e.g. arm, leg, wing, tail, stem, lobe, arch, base, etc.

1.4.3. Script conversion

- for the conversion of one script to another, the words ‘transliteration’ and ‘transcription’ are sometimes used interchangeably, but they have more restricted, and distinct, meanings in the usage we encourage
- **transcription** is “when the **phonemes** of a source language written in a dissimilar script (or not written at all) are represented more or less faithfully by the characters (letters and other graphic signs) of a dominant script” (Wellisch 1978, 18, emphasis added)
- **transliteration** is “when the **graphemes** of a source script are converted into graphemes of a target script without any regard to pronunciation and also, at least in the strictest sense, without either adding or deleting any graphemes that are not present in the source script” (ibid.)
- by the same author’s definition, **Romanisation** is “used as a neutral term to denote both methods of script conversion ... into the Roman script” (ibid., 19)
- in the case of Sanskrit, with its nearly perfect match between phonology and native writing system used to transcribe it, it takes some thinking to come up with examples of the difference between transliteration and transcription, e.g.
 - *jihvāmūliya*, *upadhmāṇiya* and regular *visarga* are differentiated in transliteration, but only *ḥ* is used in the transcription of standard Sanskrit
 - spellings such as *karmma* and *satva* are retained in transliteration, but normalized to *karma* and *sattva* in transcription

– by contrast, in situations where the match between phonology and writing system is less perfect, the distinction between the two forms of Romanisation becomes easier to make

1.4.4. Notation for transliteration and transcription

Partly for use in this guide, and partly as a reminder of the scholarly conventions that we recommend DHARMA team members adopt on the (probably rare) occasions that this will be useful or necessary, we define the use of the following brackets in the following functions:

- <...> graphemic transliteration
- /.../ phonological transcription
- [...] phonetic transcription

We presume team members will rarely have need to offer phonetic transcription, but include the square brackets (which in other contexts may bear other meanings) for completeness. We presume all team members are familiar with the distinction between phonology and phonetics, or if not have the ability to look it up on Wikipedia.

2. General Principles

2.1. Character Set and Input Method

- always use the Unicode code table (<https://www.unicode.org/standard/standard.html>),
 - never a custom/legacy encoding (i.e. one that turns into gobbledygook if you change the font to a Unicode font for the same script)
- wherever available, type using Unicode precomposed characters
 - e.g. for *ā* use the Unicode character **U+0101** Latin Small Letter A With Macron, not a combination of *a* (**U+0061** Latin Small Letter A) and *̄* (Unicode 0304 Combining Macron)
- the notation **U+####** means a Unicode character identified by the four-digit hexadecimal code **####**
- the font you use in your texts is irrelevant so long as it is Unicode-compliant
 - freely available fonts supporting all or nearly all of the special characters we require include:
 - Gentium, <https://software.sil.org/gentium/> and several other fonts by SIL
 - Google’s Noto Serif (and Sans Serif) fonts, <https://www.google.com/get/noto/>
 - several of the fonts shipped with Windows 10, e.g. Times New Roman, Tahoma, Calibri
 - several of the fonts shipped with Mac OS, e.g. Times New Roman, Arial, Calibri
 - you probably already have a favourite keyboard layout to access the special characters you need in your work
 - if not, and you are a Mac user, you may want to try the layouts Easy Unicode or ABC Extended (formerly US Extended)
 - there is, unfortunately, no readily available solution for a Windows platform, but you may be able to use and/or adapt John Smith’s keyboard layout and Word macros, available at <http://bombay.indology.info/software/fonts/induni/index.html>
- if you can access most of the characters you need via your keyboard, but there are a few that you need occasionally and cannot access, one of the following solutions may help:
 - assign a shortcut key or sequence to the inaccessible characters in your editing software
 - copy and paste the inaccessible characters from this guide each time you need one of them (or save a separate document with those characters, keep it at your fingertips, and copy-paste from that)
 - insert them from a table of available characters
 - in MS Office, use Insert Symbol
 - on Mac OS (systemwide), use the Character Table
 - use Unicode codes to enter special characters
 - in MS Office you can type the code, then press ALT + x to convert the code into the corresponding character
 - you can omit prefix U+, but using it will make certain the software recognises where the code begins, so the last characters you typed before the code will not interfere with what you want to produce
 - on Mac OS (systemwide), you need to enable Unicode Hex Input in Language Preferences
 - once you have done this, whenever you switch to this keyboard layout, you can press and hold Option while you type the character code (without the prefix U+) then release Option
- if all else fails, then consistently type one and the same particular alternative character throughout your corpus (e.g. *r* instead of *ṛ* or *š* instead of *ś*, etc.)
 - do not use that particular sign for any other purpose than representing the character you cannot type
 - make clear note of what you are doing, so your custom character can then be auto-converted to the correct one
- please note that detailed technical instructions on installing and using keyboard layouts or assigning shortcut keys are beyond the scope of this guide

2.2. Transliteration in Practice

- as Wellisch (1978, 314) points out, “there is no single ‘scientific’ system whose principles can be applied uniformly to all scripts and for all purposes ... Rather, there is a plurality of more or less justified but mutually incompatible requirements ... so that a choice must be made among those requirements that are *optimally* needed to make the system work for a particular purpose or task.” (emphasis original)
- in addition to the notion that no single Romanisation system can be applied in a practicable manner to all known scripts and languages, this implies that for actual Romanisation systems to work, they need to find an optimal point on the continuum between ideal transliteration and ideal transcription

2.2.1. Strict transliteration

- as our aim in epigraphic editions is to faithfully reflect the graphemes (characters) of the original script, the Romanisation system prescribed in this guide is very close to the transliteration end of the spectrum, and therefore we refer to it as “strict transliteration”
- the same aim, and thus the same Romanisation system, applies to diplomatic editions of single manuscripts, and for readings of specific manuscripts cited in the apparatus of a critical edition
- when strict transliteration is called for, fully prioritise transliteration over transcription except in specific cases where this guide explicitly calls for the use of Romanisation more akin to transcription (such as §2.6.4 and §3.2)
- this applies even when you are certain that a specific *akṣara* was pronounced in a way unlike that dictated by the inherent logic of the script; see §3.3.9 for some specific examples

2.2.2. Loose transliteration

- however, in other contexts, a method of Romanisation closer to the transcription end of the spectrum (which we term “loose transliteration”⁵) is acceptable and recommended, primarily in the following situations
- in the text of a critical edition of multiple manuscripts, especially where there is a mismatch between script and language (e.g. over- or underspecificity of the script for the phonemic system)
- when citing isolated words, names or passages from an inscription in a modern-language discussion
- the Romanisation scheme you use in such contexts is to be guided by your preference and the conventions of your field, and may differ from strict transliteration for instance in
 - avoiding specific representation of certain features of the writing system such as independent vowels, final consonants or the particular way a ligature is composed
 - normalisation by reducing graphic diversity in a writing system that has more characters than the phonology of the language needs, i.e. merging alternative notations of a single phoneme into one sign (that must also be a member of the larger subset of signs used in our transliteration scheme), e.g.
 - substitution of the class nasal for *anusvāra* or vice versa
 - Old Javanese *vvañ/vvañ* merged into *vvañ* (phonologically /wvaŋ/), *luraḥ/lurah* merged into *lurah* (phonologically /lurah/)
 - Old Javanese *Ṛ* interpreted as expressing the syllable *ra*, as in *səṚḥ* (in strict transliteration) corresponding to *sərah* in loose transliteration
 - disambiguation where a language uses one feature of a writing system to represent more than one phonological feature, e.g.
 - Old Sundanese *sastra*, *rahiyañ* and *ku nu reya* (even when written as *saṣṭā*, *ku nu rye* and *rahiyñ* as in the examples under §3.3.9)
 - normalisation of orthography, e.g.
 - simplification of consonants doubled in conjunction with *r* in Sanskrit

⁵ Loose transliteration is a generic term that allows for the possibility that certain non-phonological features are retained in Romanisation while others are transcribed phonologically. In many practical applications, our “loose transliteration” can be justifiably called (and has been called) simply “transcription.”

- simplification of consonants doubled at morpheme boundaries in Old Javanese (e.g., *lavann ika* simplified to *lavan ika*, *muvaḥhakan* to *muvaḥakan*, *at thana* simplified to *at hana*)
- distinction of *e/ē* and *o/ō* even if not present in the original writing

2.2.3. Shorthand

- some subsections of this guide offer methods we call “shorthand”
- these involve the use of specific characters that are not considered to be part of the DHARMA transliteration system, but are offered as an easy-to-produce alternative to
 - certain Unicode characters that are not readily accessible on many keyboards (“shorthand transliteration”)
 - certain character features that we encode in XML but which cannot be represented by our transliteration scheme (“shorthand markup”)
- the rationale of using shorthand is to ease and speed the work of encoders by permitting these easy alternatives, which can at a later stage of work be converted in bulk to the correct transliteration character, or to the correct combination of transliteration and XML markup
- our aim is to devise automated conversion at a later stage for some or all of the shorthand alternatives suggested in this Guide, but you or another project member may need to run search-and-replace (with or without regular expressions) in order to convert some shorthand items to the proper notation
- you as an encoder are also free to invent and use shorthand of your own for any feature, but for any personal shorthand, you will need to be prepared to convert it to the proper notation when your files are finalised

2.3. Transliteration Scheme

- in general, use the **ISO-15919** transliteration system for all languages written in an Indic script
 - the standard, published as a pamphlet, is accessible in the form of a pdf file in the PDF Library on Sharedocs⁶
 - Wikipedia (https://en.wikipedia.org/wiki/ISO_15919) summarises the essential features
- if you are used to IAST, this means paying attention to using *m̄*, *r̄*, *r̄̄* and *l̄* rather than *m̄*, *r*, *r̄* and *l*
- if you are used to the scheme of the Madras *Tamil Lexicon*, rest assured that it is identical to ISO-15919 on all fundamental points
- for Kannada, we will align as much as possible the guidelines on Kannada transliteration drafted by Andrew Ollett and Sarah Pierce Taylor (forthcoming), although at this stage it is unclear whether agreement can be reached on all points

2.4. Case Sensitivity

- in general principle (as per ISO-15919 Rule 8.1.1), our transliteration is case insensitive
- however, we propose to supplement ISO-15919 and – in strict transliteration – use certain uppercase letters to distinguish final consonant characters (see §3.3.1) and independent vowel characters (see §3.3.3) of the original script
 - this distinction may in some cases be redundant, but it can be particularly useful
 - where the original inscription could have used a regular *akṣara* (e.g. कृतमेतत्) but chooses instead to use a final consonant followed by an independent vowel to represent a pause for semantic or metrical segmentation (e.g. कृतम्एतत्)
 - where part of the original is not legible, and a lacuna is preceded by a consonant or followed by a vowel, this notation makes it clear to the reader whether
 - the preceding consonant is a final form or a partial *akṣara* (with an illegible vowel component)
 - the following vowel is an independent form or a partial *akṣara* (with an illegible consonant component)

⁶ <https://sharedocs.huma-num.fr/wl/?id=3y8R1K48Budcn6HjZdWcQV88xooR66kv>

- it also eliminates the need for a special disambiguation character (for which see §2.5) to distinguish vowel hiatus involving an *a* followed by an *i* or a *u* from the diphthongs *ai* and *au*
- therefore, in strict transliteration use uppercase only for these special features, and use **only lowercase** letters everywhere else, including
 - the initials of proper names, and
 - the beginnings of paragraphs, sentences, metrical units, etc.

2.4.1. A note on the use of uppercase for standalone vowels and consonants

- some of us have previously adopted the system of using a ° character before transliterated vowels and others have used the same after transliterated consonants to denote special forms for independent vowels and final consonants
- the principal investigators have agreed to discontinue using that notation, so henceforth it should not be used in XML files
 - it is also recommended that you adopt the uppercase notation in all your work including printed publications
- intellectual considerations in favour of adopting the uppercase notation include the following:
 - whereas our use of the middle dot · to transliterate explicit zero vowel markers (see §3.3.2) allows us to add markup to such markers as separate from the consonants to which they are attached, there is no such equivalence in the case of special character forms, which are more rigorously transliterated using a single Latin character than by a digraph
 - if we postulate that the ideal type of an *akṣara* is a combination of consonant(s) + vowel, then our rules mean using lowercase for normal *akṣaras*, while uppercase is used for vowels which are special by lacking a consonant, and for consonants which are special by lacking a vowel (and an explicit *virāma*)
 - uppercase letters are pre-existing special forms of Latin letters, which are easy to type on all keyboards and can be readily co-opted for our purposes as case is not used for any other purpose in ISO-15919
 - search algorithms will find text written with special forms without requiring special provisions (e.g., a search for *tam eva* will also find *taM Eva*, but fail to find *tam° eva*), whereas if only a specific orthography is desired, a case sensitive search will find only the desired string
 - using uppercase letters for special forms allows us to keep the sign ° free for the conventional use as a marker of truncation (e.g. when cutting words to be cited in a critical apparatus)

2.5. Disambiguation

- since our transliteration standard includes digraphs (e.g. *ph*, *au*), it occasionally happens that such digraphs must be distinguished from juxtapositions of the characters transliterated by the individual components of a digraph (e.g. *p* followed by *h*; *a* followed by *u*)
- in accordance with ISO-15919 (Rule 8.1.15), we use the colon (:) as a disambiguation sign where our transliteration would be ambiguous without such a sign
 - note that a disambiguation sign is not required if an editorial space or hyphen separates the two characters in question, since the transliteration is already unambiguous in this case without
- in ISO-15919, a disambiguation colon is used between vowels in hiatus to distinguish certain vowel sequences from diphthongs transliterated by the same Latin vowels
 - e.g. Sanskrit प्रउग and Prakrit चउत्थो and दइआ must be kept distinct in transliteration from प्रौग, चौत्थो and दैआ, which ISO-15919 achieves by transliterating them as *pra:uga*, *ca:uttho*, *da:iā*
 - however, our strict transliteration system⁷ provides ways of distinguishing independent vowel signs of the original script from vowel markers (see §§3.3.3 and 3.3.4), and thus we can transliterate the above words as *praUga*, *caUttho* and *daIĀ*

⁷ We recommend that in **loose transliteration** you follow the established convention of using a diaeresis (pair of dots) above the second vowel, thus प्रउग, चउत्थो and दइआ become *praüga*, *caüttho* and *daïā*.

- as a consequence, we only need a disambiguation sign to distinguish consonant + *h* combinations from aspirated consonants (e.g. *p:h* for *p* conjoined to *h* to distinguish it from the aspirate *ph*)
- accordingly, we have chosen to preserve alternate uses of the colon for some special purposes, namely to indicate the use of the *ā* marker in Indonesian scripts as an indicator of vowel length or consonant doubling (§3.3.6)

2.6. Editorial Additions for Text Analysis

- as a general rule, do not add anything to your transliteration that is not already present in the original text
- in order to handle editorial additions and alterations, you must rely on markup; see EGD §6
- however, this general rule comes with the following exceptions, which serve as a low-level editorial markup to facilitate the analysis and segmentation of a text for human readers, and which may at a later stage be converted to machine-readable XML markup

2.6.1. Editorial spaces for word segmentation

- **words** should be **separated** from one another with a space wherever Romanised transliteration allows, notwithstanding that the original inscription or a published edition, whether in Indic or Latin script, does not do so
- emphatically, **do add spaces**
 - where the end of one word and the beginning of the next word constitute a single *akṣara* in the original
 - even if such an *akṣara* involves a sandhi modification, e.g.
 - Sanskrit *tad dhi* (for *tat + hi* – space goes between *d* and *dh*)
 - Sanskrit *gacchaty eva* (space goes after the *y*)
 - Sanskrit *putrāṃḥ lakṣmīḥ* (space goes between the two *l*-s)
 - Old Javanese *tann inaku* (space goes between the *-nn* and the *i*-)
 - Tamil *arit' enru* (for *aritu + enru*; see also §2.6.4 for elision of overshoot *u* in Tamil)
 - including non-standard sandhi and orthographic practice, e.g.
 - nasals used where standard orthography would employ an *anusvāra*, e.g. Sanskrit *uktañ ca* or *śaraṇaṅ gataḥ*
 - Sanskrit *dīnārair ddaśabhiḥ*
 - Old Javanese *darpaṇa ryy avakta*
 - before an *avagraha*, unless it occurs within a compound
 - thus use, e.g. *so 'bhūt* and not *so'bhūt* for separate words, but use e.g. *saro'nte* or *saro-'nte* in compound, depending on whether or not you use hyphens for compound segmentation
 - in close-knit structures such as *atha vā, kiṃ ca* and *kiṃ tu* (even if spelt *kiñ ca* and *kin tu*), *tad yathā*; including grammaticalised structures such as
 - Sanskrit periphrastic perfects, e.g. *varayām cakāra* (especially since other words may intrude inside such a construction, e.g. *saṃraṃjayām ca prakṛtīr babhūva*)
 - Sanskrit past tense formed with imperfect and *sma*, e.g. *samādiśati sma*
 - Sanskrit formations with *-sāt* prefixed to a verb such as *brāhmaṇasād gatāḥ*
 - Sanskrit prepositions such as *ā samudrāt, anu gaṅgām*
 - in repetitions of Sanskrit inflected pronouns and nouns (*āmreḍita*) expressing a generalised or distributive meaning, e.g. *yasya yasya, dine dine*
- **do not**, however, use spaces (nor hyphens) to separate
 - successive words where the final vowel of the first and the initial vowel of the second are fused in vowel sandhi, e.g.
 - *tasyāyam* stays as is, though *so yam* is separated
 - *gacchatīva* stays as is, though *gacchaty eva* is separated
 - Tamil enclitic particles (e.g. *ē, ō*) and forms of the verb *āku-tal* (e.g. *āṇa, āy, āka*) when used adverbially
 - Old Javanese enclitic pronominal suffixes (*-(n)ku* etc.), possessive constructions built with the linker *-ni* (*-nikaṅ*, etc.); definite article *-ni*; conjunction *-n*

- for Sanskrit close-knit structures borrowed into other languages, follow the spelling with or without space (generally the latter) of the relevant dictionaries, if there are any
 - e.g. Old Javanese *kimuta*, Old Cam *kintu*
- in sub-standard Sanskrit, strings of words without case endings but apparently intended as nominatives should preferably be spaced instead of being treated as compounds (e.g. *dvandva*), unless the latter in fact facilitates interpretation, e.g.
 - *lamvoṣṭha dedamita mahādeva guṇṭhaka ity evam-ādibhyo*
 - *samrāṭ vākātakānām mahārāja śrī-pravarasenasya*

2.6.2. Editorial hyphenation

- editorial hyphens may be optionally added for the following purposes
 - **segmentation of compounds** in Sanskrit and other compound-heavy languages
 - such segmentation need not be exhaustive
 - feel free to hyphenate only long or difficult compounds and leave others intact
 - as a rule, do not use hyphens within proper names (e.g. *śrī-viṣṇuvarddhana-mahārājasya*, not *śrī-viṣṇu-varddhana-mahārājasya*)
 - but do feel free to use hyphens when part of a compound name may be perceived as an epithet rather than an essential part of the name (e.g. *śrī-jayasimha-vallabha-mahārājasya*, where the name Jayasimha also occurs without the epithet *vallabha*)
 - in Sanskrit, keep in mind that secondary derivatives of compound nouns are not themselves compounds and should therefore not be hyphenated, e.g.
 - although derived from the compound *catur-varṇa*, *cāturvarṇya* is not a compound and should not be divided up into the members **cātur* and **varṇya*
 - Sanskrit compounds where a verb is combined with a nominal stem with the ending *-ī* (or *-ū*) should for this purpose be regarded as compounds and may be hyphenated when this is deemed helpful for the reader; specifically, we recommend that you
 - preferably hyphenate such compounds when they are not themselves members of compounds and/or when their nominal member is itself a compound (in which case preferably avoid hyphenation within that member), e.g. *brahmadeyī-kṛtya*
 - preferably avoid hyphenating such compounds when they are themselves members of longer compounds and/or when their nominal member is a simple word, e.g. *aśvamedhāvabhṛtha-snāna-pavitrīkṛta-śirasām*
 - in the case of Old Javanese, consider that reduplicated expressions are always compounds, whereas close-knit structures consisting of two different elements only become compounds if any morphological derivation takes place
 - **sandhi analysis** when hyphens are conventionally used for this purpose in your field, specifically:
 - epenthesis in Tamil may be indicated by joining the added letter to the preceding word with a hyphen (see the examples below)
 - non-standard Sanskrit sandhi involving an epenthetic *m*, *s* or *r* may be indicated in the same way, e.g. *mleccha-rājye-m apūjitaḥ*; *pañca-s-trimśottaratame*
- as with editorial spacing, feel free to add hyphens between transliterated characters that belong to a single *aḥsara* of the original, but do not use a hyphen at points where the final and initial vowels of two successive compound members are fused in sandhi
- some examples of Tamil hyphenation:
 - *tiru-makaḷ* (திருமகல் *tiru+makaḷ*)
 - *koṇṭ-āṭu* (கொண்டாடு *koṇṭu+āṭu*)
 - *I-p-peruñ-kōyil* (இப்பெருங்கோயில் *i+peruñ+kōyil*)
 - *tiru-mēṇi-y āṭa* (திருமேனியாட *tiru+mēṇi āṭa*)
- some examples of Old Javanese hyphenation:
 - *vulu-vulu*
 - *tahi tikus > manahi-tikusa*

- no fully satisfactory hyphenation point can be identified when an *h*-initial word is the second member of a compound whose first member ends in an unvoiced stop, and an aspirated consonant *akṣara* is employed to represent the final phoneme of the initial member of the compound plus the phoneme /h/ of the second member, and consonant gemination at morpheme boundaries occurs as well: e.g., *bvat haji* > *buAtthajyanya*, to be hyphenated, for lack of a better solution, as *buAt-thajyanya*
- **do not use hyphens** for any other purpose, e.g. to show that a word has been broken into two parts by the end of an inscribed line
 - this should be noted in markup (see EGD §3.2.4)
 - if you are not adding any markup, please use the character - (U+00AC Not Sign; do not use a hyphen), which will be auto-converted into the proper markup
- if you use hyphens for editorial compound analysis, and
 - a **physical line break** coincides with such a hyphen, then
 - first encode the physical line break as one inside a word (as per EGD §3.2.4 or with the shorthand -)
 - then put the editorial hyphen at the beginning of the new line
 - a **verse line break** coincides with such a hyphen, then
 - first encode the verse line break as one inside a word (as per EGD §2.3.6)
 - then put the editorial hyphen at the beginning of the new line

2.6.3. Representation of *avagraha*

- use the transliteration ' (right single quote) to represent any original *avagraha* in your text
 - alternatively, you may use the shorthand ' (plain apostrophe) if this is easier for you to type
- the inscriptions we work with very rarely use an actual *avagraha* sign, but *avagrahas* may be optionally supplied by the editor
 - this is recommended especially in cases where the text would be meaningful (and even contradictory in meaning) both with and without an *avagraha*
 - e.g. the inscribed sequence *sohataḥ* may stand for *so hataḥ* or *'hataḥ*, so if you interpret the text as the latter, then supply an *avagraha* to make this clear
 - if you feel that ambiguity of this kind is a deliberate poetic device (bitextuality, *śleṣa*), then the decision whether or not to supply an *avagraha* should be based on what you consider to be the *prima facie* meaning of the text
 - the recommendation of supplying *avagrahas* applies likewise to words in compound (and regardless of whether you use hyphens for compound segmentation or not), e.g. *yaśo'mṛtam* or *yaśo-'mṛtam*; *saro'nte* or *saro-'nte*
- since most of the *avagrahas* in our texts will be supplied, any and all *avagrahas* in an electronic edition will be assumed to be supplied by the editor, and the required markup signifying this (for which see EGD §6.3.7) will be added automatically
 - in the exceptional cases where there is an original *avagraha* in your texts, use the shorthand '! or '!' (right single quote or plain apostrophe followed by an exclamation mark)
 - the automatic conversion routine will not add supplied markup to these, but will remove the exclamation marks after adding XML tags to all supplied *avagrahas*
- in order for automatic conversion to be possible, an apostrophe representing a supplied *avagraha* must never be followed by a space, in order to distinguish it from the apostrophe used to represent elision in Tamil (q.v. §2.6.4), which is always followed by a space
 - such a distinction is important because whereas the supplying of the Sanskrit *avagraha* is an act of normalisation, namely projecting modern orthographic practice (e.g. सो ऽहतः) onto an original text in which this practice was not (or not widely) used, the use of an apostrophe in the transliteration of Tamil is an act of linguistic analysis, considered an integral part of our transliteration scheme
 - a Sanskrit *avagraha* must normally be preceded by a space (§2.6.1), but it will be preceded by an alphabetic character or a hyphen when it appears within a compound; only the following space is relevant to automatic conversion

2.6.4. Representation of elided overshoot final *u* in Tamil

- in the transliteration of Tamil text, use an apostrophe followed by a space to represent the elided overshoot *u* at the end of an independent word, e.g.
 - *arit' enru* (அரிதென்று for *aritu + enru*)
- but do not use an apostrophe for the elided overshoot *u* inside a lexicalised compound, e.g.
 - *koṇṭ-ātu* (for *koṇṭātu*)
- note that an apostrophe used for this purpose must always be followed by a space (and not be preceded by one), in order to distinguish it from the apostrophe used to represent Sanskrit *avagraha* (q.v. §2.6.3)
 - such a distinction is important because these apostrophes are used in the transliteration of Tamil as a feature of linguistic analysis (in a way similar to our use of editorial hyphens, §2.6.2): they are understood to be integral parts of our transliteration system which do not correspond to anything graphically present in the original, whereas *avagrahas* in Sanskrit could conceivably be present in the original, and when they are not, they are supplied by the editor for the sake of normalisation

3. Alphabetic Characters

3.1. Some Special Characters

- most of the characters below are covered by ISO-15919, but are specifically mentioned here because their transliteration may not be self-evident to all of us
 - ! transliterations not covered by ISO-15919 will be marked in this section by an initial exclamation mark
- **vocalic *r* and *l***
 - these are not available in Unicode as pre-composed characters, so to create them, you may need to enter an *r* or *l* as applicable, followed by ◌̣ (U+0325 Combining Ring Below) and, if needed, by ◌̄ (U+0304 Combining Macron) in this order
 - alternatively, since none of the languages we work with require the use of *r* to represent a consonant, you may optionally use the shorthand *ṛ* and *ḷ*, which will later be automatically converted to *r̄* and *l̄* in your files
 - note, however, that this does not apply to *l*, because some of the epigraphic fields in which we work do require a distinction between consonant *l* and vowel *l*
- ***anunāsika/candrabindu***
 - *ṃ* (this character is not available as a precomposed glyph, so it must be composed of a regular *m* and a ◌̣ sign: U+0310 Combining Candrabindu)
 - use **only** if distinguished in the original script from *anusvāra*
 - but, conversely, **always** make the distinction in transliteration if the distinction is made in the original
 - *candrabindu* signs enlarged and embellished for ornamentation do not receive a different treatment in transliteration
 - only add the Candrabindu sign to *m* (i.e. avoid using *tāḷ lakṣmīm* and write *tāṃl lakṣmīm* instead)
- ***upadhmanīya*** (if distinguished in the script from *visarga*)
 - *ḥ* (U+1E2B Latin Small Letter H with Breve Below)
- ***jihvāmūliya*** (if distinguished in the script from *visarga*)
 - *ḥ̣* (U+1E96 Latin Small Letter H with Line Below)
- **Tamil *āyṭam*, ೀ**
 - *ḷ* (U+1E35 Latin Small Letter K with Line Below)
- **retroflex lateral**, Tamil ள Kannada/Telugu ೞ
 - *ḷ* (U+1E37 Latin Small Letter L with Dot Below)
- **alveolar trill/stop**, Tamil ள Kannada/Telugu ೞ
 - *ṛ* (U+1E5F Latin Small Letter R with Line Below)
- **retroflex approximant / frictionless continuant**, Tamil ೞ Kannada/Telugu ೞ
 - *ḷ̣* (U+1E3B Latin Small Letter L with Line Below)
- ! **Cam *anusvāra-candra***
 - *ṃ* (this character is not available as a precomposed glyph, so it must be composed of a regular *m* and a ◌̣ sign, U+0303 “Combining Tilde”)
- ! **Javanese/Balinese *pepet*** (expressing the vowel schwa)
 - short, ə (U+0259 Latin Small Letter Schwa); uppercase Ə (U+018F Latin Capital Letter Schwa)
 - you may use the shorthand *ě* instead of ə if it is easier for you; since *ě* is not used for any other purpose in our transliteration, it can be automatically converted to ə
 - long, ǣ: (with length-mark represented by a colon as per §3.3.6) in strict transliteration
 - ǣ̄ in loose transliteration (not available as a precomposed character: add U+0304 Combining Macron to the plain character)
- ! **Khmer (and Mon-Burmese) glottal stop**
 - *q* (the Latin letter q)
 - see also §3.3.4 about the representation of independent vowels involving this character component
- ! **special signs for Mon and Pyu:**

- barred/dotted variant of *b*
 - *ḃ* (U+1E05 Latin Small Letter B with Dot Below)
- *akṣaras* with underdot
 - *ṁ* (U+1E43 Latin Small Letter M with Dot Below)

3.2. Long and Short *e* and *o*

- when transliterating a language that does not make a distinction between long and short *e* and *o*, use these Latin characters without a macron
 - this corresponds to Option 9.1 of the ISO15919 standard, applicable to languages that do not make a distinction between the phonemes *e/ē* and *o/ō*
- however, for Dravidian **languages that distinguish long and short *e* and *o***, you have the option to record that distinction even if it is not present in the script you are working with
 - in this case, transcribe long vowels as *ē/ō* even in strict transliteration
 - these characters will be treated as shorthand, subsequently to be automatically expanded with XML markup signifying that *e* or *o* was in fact inscribed, but the spelling has been normalised by the editor
 - that is to say, the palaeographically primary generic vowel marker, e.g. that in **കെ** *ke*, **കോ** *ko*, may represent either a short or a long vowel; when it represents a long vowel, this will be shown as an editorial normalisation, e.g. to **കേ** *kē*, **കോ** *kō*
- should your inscription (or manuscript) explicitly distinguish between short and long *e/o*, please contact us to devise a solution for handling this

3.3. Special Glyph Forms and Compositions

- ideally, transliteration would not be concerned with what allograph is used in a particular instance to represent a particular grapheme
 - however, we find that it may be important for our research interests to preserve in the transliterated text some alternative ways of representing the same character or character combination
- for this reason, in strict transliteration we shall employ some mandatory distinctions set out in the following subsections
- certain further distinctions set out in the following subsections may be optionally made using markup or a shorthand notation that will be auto-converted to markup
- other potentially interesting allographs – for instance the use of two alternative glyphs within the same inscription for the same simplex character, or different ways in which a vowel marker is attached to a consonant – will need to be described in your metadata, and will not be directly represented in the transliteration or the markup⁸

3.3.1. Final consonants as special simplex characters

- special character forms representing consonants without a vowel (called *halanta* consonants in Sanskrit) are typically a miniature and/or subscript rendering of a simplex consonant *akṣara*
- such special final forms shall be mandatorily transliterated as a corresponding uppercase Latin consonant, e.g. *T*
- the criterion by which to distinguish special final forms from complex characters involving a zero vowel marker (§3.3.2) is the use of a glyph distinct (in size, shape and/or position relative to the baseline) from the regular simplex character employed for that consonant with an inherent *a*
 - if this criterion is met, then the character in question should be transliterated with an uppercase consonant even if the special form includes a component that may be perceived as a zero vowel marker, including in particular the horizontal dash above a miniature consonant sign in Indian inscriptions, which may be viewed as a proto-*virāma*, but which we treat as part of the special consonant form, not as an explicit zero vowel marker

⁸ This is a conscious decision of the authors of this Guide, who consider that we need to impose a limit on the granularity of our representation of potentially interesting phenomena. However, it is possible to use sub-*akṣara* markup (EGD §4.1.2) to encode the relative positions of certain character components, if you consider it necessary to do so.

3.3.2. Final consonants as complex characters involving a zero vowel marker

- complex characters involving a regular simplex form and an explicit zero vowel marker (*virāma*, *pulli*, *patén/pangkon*, etc.; also referred to as “vowel killer”) shall be mandatorily transliterated as follows
 - type the character `·` (U+00B7 Middle Dot) after the Latin consonant, e.g. *t·*
 - if you have difficulty typing this sign, optionally use an asterisk `*` as shorthand;⁹ this will be replaced later on with the middle dot
- use the same method to represent a **Tamil *pulli*** that is explicitly present in your original (e.g. *t·ta* to transliterate த்த)
- where *pulli* is not present in an inscription but is to be understood implicitly, simply type the transliterated consonant cluster without any additional characters (e.g. *tta* to transliterate த்த understood as த்த)
- we may at a later point decide to automatically convert such transliterations into markup signifying that a *pulli* has been supplied by the editor, but for the time being our default assumption is that any consonant cluster in transliterated Tamil involves an implicit *pulli*
- note that where an actual ligature occurs in Tamil script, this must be treated as unusual *akṣara* composition, for which see §3.3.8
- representing zero vowel markers by a separate character in the transliteration has the added advantage of allowing the application markup to this sign, e.g. to tag it as unclear, restored or supplied

3.3.3. Independent vowels as special simplex characters

- if the original script employs a distinct character for vowel-only *akṣaras* (initial vowels and vowels in hiatus), these shall be mandatorily transliterated as follows
 - type a corresponding uppercase Latin consonant, e.g. *A*
 - thus, इति becomes *Iti*, whereas कृतमिति becomes *kṛtam iti*
 - for the independent forms of the diphthongs *ai* and *au*, capitalise only the first character of the digraph in your transliteration, i.e. use *Ai* and *Au* (whereas *AI* and *AU* would transliterate अइ and अउ, should these combinations occur)

3.3.4. Independent vowels as complex characters involving a “vowel support”

- in some of the writing traditions relevant to DHARMA, the independent vowels of the Indic syllabary, and particularly the independent vowel that we normally transliterate as *A*, have come to be used in manners that would not be expected in India and that are not foreseen by ISO-15919.
 - the character used in such cases as a “vowel support” may represent a glottal stop, or be only a zero consonant sign, serving no other function than to permit the notation of a vowel
 - in some writing systems, we see over time that a consonant character that is not needed to express any consonant phoneme gets redeployed as vowel support (e.g., the *h* in Modern Javanese script where the Sanskrit loanword *eka* is spelt *heka*)
 - in others, inherited “independent vowel” characters gain the ability to behave as consonant characters, i.e., to bear vowel markers and to enter into ligatures
 - this systemic change is complete in Khmer, where 𑀓 functions fully as a consonant *akṣara*, but has been carried through to varying degrees in the writing traditions of Java, Bali and Lombok¹⁰
- the fact that “vowel support” characters can enter into combinations not foreseen by ISO-15919 may cause problems of ambiguity in transliteration in ISO-15919, or it may simply be impossible to transliterate them without adapting the system

⁹ Note that the asterisk is also used in our scheme to distinguish special forms of *anusvāra*, §3.3.11.

¹⁰ According to Ida Bagus Komang Sudarma (personal communication, 16 Aug. 2019), in Sasak writing 𑀓 can be combined with a *pasangan* consonant, e.g. 𑀓𑀧 *qhi* and 𑀓𑀧 *qhu*, but cannot itself become a *pasangan*, while in Balinese writing neither possibility exists.

- in the remainder of this section, we will be concerned only with “independent vowel” characters that are used as “vowel supports”, and whose functional equivalence to consonant characters will be made transparent in our system either by their position (immediately after a consonant character, without intervening inherent vowel *a*) or by our use of the dedicated character *q*
- in short,
 - **when the “independent vowel” character that we would otherwise transliterate as *A* is employed in the original script as a “vowel support”, i.e.,**
 - (1) when it enters into a ligature with another consonant,
 - (2) when a vowel marker is attached to it, or
 - (3) when it enters into a ligature with another consonant and a vowel marker is attached to the ligature,
 - **then mandatorily transliterate the “vowel support” with the letter *q* followed by the applicable (lowercase) Latin vowel**
- in all other situations, even if it is clear that they express a consonant phoneme, transliterate “independent vowel” characters in the manner prescribed in §3.3.3
- here is an overview of how to transliterate relevant cases from Khmer script
 - 1. Simplex “independent vowel” characters
 - អ *A*, អា *Ā*, ត *I*, ព្រំ *Ī*, ឧ *U*, ឌី *Ū*, ង *E*, ច្ន *Ai*, ឌី *O*, ឌី *Au*
 - 2. អ used as vowel support in syllable-initial position
 - អី *qi*, អី *qī*, អុ *qu*, អុ *qū*, អេ *qe*, អៃ *qai*, អោ *qo*, អៅ *qau*
 - 3. អ in the function of vowel support (2) and simplex equivalents (1)
 - អេត *qeta* — ឌត *Etta*
 - 4. អ entering into a ligature as vowel support in syllable-initial position
 - អ្នក *qnaka*
 - 5. អ entering into a ligature as vowel support in post-consonantal position
 - ផ្នក *phqaka*
 - ល្អិត *lqita* as opposed to ល្អត *lita*
 - ប្រឌ *pqūna* as opposed to ផ្នឌ *phUna*
 - ក្រក *kqeka* as opposed to ក្រក *kEka* and ក្រក *kqaeka*¹¹

The following table repeats some of the same examples but adds potential equivalents in Balinese glyphs, and shows that the same identically transliterated sequence may not lead to the same transcribed sequence between one writing system and another, because at the level of transcription, we must start to take into account the sounds that are being expressed. The last column shows the romanization that you may use, for instance, in commentary and translation, when you wish to cite words as words, not as transliterated strings of characters.

¹¹ The example is taken from Modern Khmer. The combination *ae* is not included in the DHARMA system because it does not exist in Old Khmer, but it does not seem to cause any problems with our system.

combination	Khmer glyph	Balinese glyph	DHARMA transliteration	IPA transcription	DHARMA transcription (lower and upper case)
A	អ	ᬓ	A	Khmer /ʔa/ Bal. /a/	qa 'a or qA 'A a A
A with e marker	ᬓេ	ᬓᬞ	qe	Khmer /ʔe/ Bal. /e/	qe 'e or qE 'E e E
A with u marker	អ៊	ᬓᬞ	qu	Khmer /ʔu/ Bal. /u/	qu 'u or qU 'U u U
A with i marker	អិ	ᬓិ	qi	Khmer /ʔi/ Bal. /i/	qu 'u or qU 'U u U
A with e and ā markers	ᬓេអ	ᬓᬞᬞ	qo	Khmer /ʔo/ Bal. /o/	qo 'o or qO 'o o O

- finally, note that when you are representing a Sanskrit name or loanword from an Old Khmer text in transcription, you should transcribe it as you would if the context were fully Sanskrit
 - e.g., Amoghapura can remain Amoghapura, notwithstanding the facts
 - (1) that this name would have been pronounced by Khmer speakers with an initial glottal stop and
 - (2) that you would transcribe the same initial syllable as 'a/qa or 'A/qA if it stood inside or at the beginning of a Khmer word or name

3.3.5. Multiple vowel markers within an *akṣara*

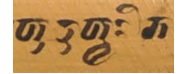
- multiple vowel markers may be used deliberately in the original
 - to represent a particular phoneme or modification (§3.3.6)
 - to mark segments as deleted (this belongs in the domain of markup, not that of transliteration; see EGD §4.5.1)
 - if you encounter multiple vowel markers that appear to be used deliberately for a purpose other than the above, please contact us to discuss how best to represent them
- other appearances of multiple vowel marks are likely to be cases where the scribe erroneously engraved more than one explicit vowel mark, neither of which appears to be deliberately cancelled
 - if one of these vowels is expected in the context and the other is not, it is acceptable and preferable to encode this as a premodern correction as per EGD §4.5.3
 - in other cases, which we expect to be very rare, transliterate all vowels in an order you deem suitable
 - the fact that the transliterated vowels are lowercase indicates in our system that none of them are independent vowel *akṣaras* (cf. §2.5)
 - nonetheless, the unusual fact that multiple vowel markers are present in a single *akṣara* may optionally be made explicit using an = (equals) sign between the transliterated vowels belonging to a single *akṣara*, e.g. *du=ā* for the glyph shown on the right (which is probably an engraving mistake for *dhā*)



3.3.6. Repurposed vowel markers

- for the **ā marker** (Javanese *tarung*, Balinese *tedong*) used as a marker of vowel length or consonant doubling in Indonesian texts:
 - when used in conjunction with another vowel marker to transform the latter into a long vowel, mandatorily type a colon (:) after the short vowel to transliterate the length marker
 - when representing a doubling of the consonant component of the *akṣara* to which it is attached, mandatorily transliterate this by typing a colon (:) after the transliterated consonant to which the doubling pertains, even if it is not adjacent to the length marker in the original script, e.g. Old Sundanese
 - **ᬓᬞᬞᬞ** (pronounce /gənəp pipitu/, “fully seven”) is to be transliterated as *gnəp:ipitu*

– the image on the right (*/turut vahaṇa/*, “You should follow his behaviour”) is to be transliterated as *turut:vahaṇa*



– for the **vowel markers *u/ū* and *i*** used together to represent a particular phoneme in Khmer, Burmese and Mon (as in the Khmer character shown in the image):



– mandatorily transliterate the vocalisation as *ui* or *ūi*

– however, the deliberate use of *u* and *i* markers in conjunction to signify deletion belongs in the domain of markup (see EGD §4.5.1), not that of transliteration

3.3.7. Short vowel written where a corresponding long vowel is expected

– where a short vowel is written in place of an otherwise identical long vowel, optionally add a breve to the transliterated short vowel in order to highlight the fact that the short vowel is not an editorial mistake

– i.e. use *ă*, *ĭ* or *ŭ* when *a*, *i* or *u* is used for expected *ā*, *ī* or *ū*

– this option is especially recommended for Sanskrit loanwords in Indonesian vernacular documents, following Damais (1955, 15), but need not be applied in such documents if notation of vowel length is absent altogether from their orthography

– this shorthand notation will be converted to XML markup involving the tag `<orig>` as per EGD §6.3.7

3.3.8. Unusually composed complex characters

– in order to highlight certain formations that deviate from the standard glyph composition for any particular language and writing system, our transliteration scheme permits the use of the dedicated character = (equals sign), in the specific cases set out below

– this notation is optional, but if you do employ it anywhere within an edition, please attempt to use it consistently throughout that edition wherever applicable

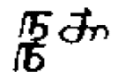
– the = sign will be ignored by search and processing software, but serve as a marker that something strange is going on in the text here, and may be used as a starting point for future analysis or harvesting of such cases

– should you need to add an editorial space or hyphen between such characters, put it *after* the = sign

– e.g. *...k=ka...* (for a single *akṣara* at a word boundary) and *...k=-ka...* (for a single *akṣara* at the boundary between two elements of a compound word)

– this notation will be auto-converted to markup (EGD §4.1.1)

– **where a Tamil text written in Tamil script employs a ligature** such as *nna* and *kka* in the image



– use an = sign between the transliterated consonants to distinguish the ligature from the script's default method of writing conjunct consonants as two glyphs with an explicit or implicit zero vowel marker (for which see also 3.3.2), e.g.

– *n=na* as distinct from *n·na* and *nna*

– *k=ka* as distinct from *k·ka* and *kka*

– it is **strongly recommended** that whenever feasible, you should make Tamil ligatures explicit in this way

– however, never add an = sign where ligatures are a writing system's default method of representing conjunct consonants (including Tamil written in Grantha)

– **where an Indonesian text employs the superscript *r* marker** (*repha*, *layar*, *surang*) in two modes,

– namely

– the “Indian” mode, i.e. to be read before the consonant it is attached to, as in *ਸਰ੍ਹ* *sarva*; and

– the “Indonesian” mode, i.e. to be read after the rest of the *akṣara* it is attached to, as in *ਸਰ* *samar*

– then this fact must be noted in your commentary to the text, including a specification of which mode is the default (dominant) one for that text

– in addition, you may optionally use an = sign between the transliterated *r* and the other characters transliterating the same *akṣara* in instances of the non-default mode, i.e.


– the representation of the “Indonesian” (versus “Indian”) positioning of the *r* marker is handled via markup

4. Non-alphabetic Characters

4.1. Numerals

- numbers written in **decimal place-value notation** in the original shall be transliterated straightforwardly (e.g. 876), since each transliterated character corresponds here to one original character
- however, original numeral characters which cannot be represented in transliteration by a single character (i.e. an Arabic digit for integers between 0 and 9, or a common vulgar fraction sign as per §4.1.2 below) require markup to indicate the fact that several transliterated characters together represent a single original character
 - this markup is ideally in the form of XML tags as per EGD §4.2.2
 - but to simplify your work, especially if you are creating an e-text that will not (or not immediately) be marked up in XML, you may take advantage of the following shorthand markup method, which will be automatically converted to the proper XML markup¹⁴
- for integers 10 and above, written as a single original character,
 - type a + sign after each transliterated number sequence of two or more Arabic numerals that represents a single numeral character in the original¹⁵
 - arguably, most Indic numerals in the 100s range could be viewed as combinations of several characters rather than as a single character, but we foresee no useful purpose that such a complex distinction could serve and therefore treat all these Indic numerals as single characters (with distinguishable components)
 - for example:
 - 10+ means “10” written as a character for 10, e.g. Brahmi α
 - 10+ 2 means “12” written as a character for 10 followed by one for 2, e.g. Brahmi $\alpha =$
 - 80+ 10+ means “90” written as a character for 80 followed by one for 10, e.g. Brahmi $\Phi \alpha$
 - 300+ 50+ 2 means “352” written as a character for 300, one for 50, and one for 2
 - 1000+ 8 100+ 3 10+ means “1830” written in Tamil notation as 1000 (plus) 8 (times) 100 (plus) 3 (times) 10
 - note that though this shorthand notation resembles the notation of older printed publications such as *Epigraphia Indica*, which used a + sign to indicate numerals inscribed in an additive system, in our shorthand + must mark the end of every sequence of two or more Arabic numerals that transliterate a single numeral character in the original
 - thus, in our shorthand a final + sign is required in cases such as the following
 - 10+ (rather than 10) – “10” written as a character for 10
 - 300+50+ (rather than 300+50) – “350” written a character for 300 and one for 50

4.1.1. Numbers denoted by bars

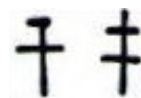
- to transliterate numerals represented in Cambodian inscriptions by bars (*daṅḍa*) instead of numeral characters (as in the image, showing the number 3):
- type as many I characters as there are bars in the original (NB: uppercase i characters, not vertical bars |)
- type a + sign after the last I as shorthand markup
 - note that unlike regular numerals, the + sign must be used in this case even after a single I representing the numeral 1
 - this shorthand will be automatically converted to markup indicating that these characters are not alphabetic and constitute a single meaningful character

¹⁴ This shorthand may be used in place of the tagging of numeral characters as per EGD §4.2.2, but it does not replace the semantic markup for the value of numerals (EGD §7.1), which is mandatory in XML files for all numerals including those inscribed in the place-value system.

¹⁵ Keep in mind, however, numerals 1 to 10 do not need this + sign even if they are used in an additive system (in combination with signs for tens or hundreds), since these are represented by a single transliteration character.

4.1.2. Fractions

- for any fractional numbers represented in the original by a single character, use one of the following Unicode characters:
 - ½ (U+00BD Vulgar Fraction One Half)
 - ⅓ (U+2153 Vulgar Fraction One Third)
 - ⅔ (U+2154 Vulgar Fraction Two Thirds)
 - ¼ (U+00BC Vulgar Fraction One Quarter)
 - ¾ (U+00BE Vulgar Fraction Three Quarters)
- should you encounter fraction signs not covered by the above,¹⁶ transliterate them as a common fraction using a slash and add a + sign after the denominator as shorthand markup
 - e.g. 1/8+ to transliterate a numeral sign meaning “one eighth”
 - this shorthand will be automatically converted to markup as per EGD §4.2.2
- the Khmer fraction sign in the shape of a cross (with a single or a double bar, see the images) shall always be transliterated as ½ (U+00BD Vulgar Fraction One Half)



4.2. Symbols

- in the terms of this Guide, “**symbol**” is an umbrella term for all non-alphanumeric characters
- we consider the diversity of non-alphanumeric symbols used in inscriptions to deserve preservation and investigation, but acknowledge the challenges of representing them in a machine-actionable scheme of manageable complexity, and therefore provide a combination of methods for handling them at different levels:
 - 1. at the level of transliteration, symbols of a definite function (punctuation marks and space fillers) shall be represented by a dedicated character, but symbols whose function is not so straightforward (“miscellaneous symbols”) shall have no representation on this level, only on that of XML markup
 - 2. at the level of XML encoding, all symbols shall be encoded with a relatively simple classification of their shapes, for which see EGD §4.2.3
 - 3. at the level of human-readable metadata, symbols may (and generally should) be described in any detail that you deem appropriate, in the handDesc section of the TEI header (EGD §11.2.1)
- as an intermediate step between levels 1 and 2 above, the following subsections recommend the use of **shorthand markup** for certain symbols
 - we may opt later on to devise automatic conversion of this shorthand to XML markup, but it seems likely at present that we will not do so
 - thus, if you use the shorthand suggested here, please be prepared that you may have to replace it with XML markup yourself, and keep notes with a more accurate description of your punctuation marks than that afforded by the shorthand scheme

4.2.1. Punctuation marks

- in the terms of this Guide, “**punctuation mark**” is used in a sense restricted to symbols employed in the original for syntactic or metrical segmentation into relatively small units, similar in function to a modern comma, full stop, question mark, exclamation mark, colon or semicolon
- the scope of punctuation marks **primarily includes** symbols
 - whose shape is derived from the small and originally simple non-figural marks used widely for punctuation in our scripts, e.g. vertical bars, dots, circles and dashes
 - whose primary function is to segment the text into sentences, clauses, list items or metrical units
 - which occur repeatedly in the body of a single text
- the scope of punctuation marks **does not normally include** symbols
 - whose shape is figural or complexly ornamental, and is not derived from one of the simple basic shapes used for punctuation

¹⁶ Although further “vulgar fraction” signs are available in the Unicode code table, they are absent from many fonts and not easily accessible via the keyboard. We have therefore chosen to use single Unicode characters only for an arbitrarily delimited set of common fractions.

- whose primary function is to mark the beginning or end of an entire inscription or a major section thereof, and/or to express auspiciousness
- which occur only once per text or once per major section of text
- **original punctuation** must always be transliterated, but you should **never add punctuation marks** not already present in the original
 - editorial punctuation may, however, be supplied using XML markup, see EGD §6.3.6
- as outlined in the introduction to this section (§4.2), punctuation marks shall be represented
 - 1. at the level of transliteration, by the dedicated character . (full stop, period), which by our convention shall be understood as an abstract punctuation mark without any assertion as to its physical appearance
 - 2. at the level of XML encoding, with a relatively simple classification of their shapes
 - 3. at the level of human-readable metadata, in additional human-readable detail
- as an intermediate step between levels 1 and 2 above, we recommend the use of shorthand markup for the basic forms of common punctuation characters, as follows:
 - all of the following shorthand characters should be followed by a space in transliteration, but not preceded by one
 - | (U+007C Vertical Line): for signs comprised of a single plain vertical bar (corresponding to the symbol token “danda”)
 - when transliterating two or more iterations of single vertical lines, make sure you add a space between them to differentiate them from double *daṅḍas*
 - || (U+007C Vertical Line, twice): for signs comprised of a double plain vertical bar (corresponding to the symbol token “ddanda”)
 - / (regular slash): for signs comprised of a single vertical bar with a hook, crossbar or ornamental addition (corresponding to the symbol token “dandaOrnate”)
 - // (two regular slashes): for signs comprised of a double vertical bar with a hook, crossbar or ornamental addition (corresponding to the symbol token “ddandaOrnate”)
 - , (comma): for short, predominantly vertical and often curved strokes normally floating at or above median height, including half-sized *daṅḍas* and the raised comma-like sign that is the basic punctuation sign on Java and Bali (modern Balinese *^*) (corresponding to the symbol token “comma”)
 - ~ (U+2013 Tilde Operator): for signs comprised of a single horizontal dash, plain or with ornamentation (corresponding to the symbol token “dash”)
 - @ (“at” sign) for any punctuation mark that does not easily fall under any of the above categories (such as more or less complex dots and circles)
 - the transformation of this character into XML markup will definitely not be automated and will have to be handled by you manually if you use this shorthand

4.2.2. Space filler signs

- in the terms of this Guide, “**space filler**” is used in a sense restricted to symbols whose function is clearly and unambiguously to fill up space in a line to the binding-hole or margin
 - symbols that do not meet this functional criterion shall be encoded as miscellaneous symbols even if they are identical in visual appearance to symbols used as space fillers elsewhere in the document or the corpus
- as outlined in the introduction to this section (§4.2), space fillers shall be represented
 - 1. at the level of transliteration, by the dedicated character § (U+00A7 Section Sign), which by our convention shall be understood as an abstract space filler without any assertion as to its physical appearance
 - 2. at the level of XML encoding, with a relatively simple classification of their shapes
 - 3. at the level of human-readable metadata, in additional human-readable detail
- we recommend the use of the following shorthand markup as an intermediate step between levels 1 and 2 above

- since the use of space fillers is most common within our corpus in Javanese documents, a § character without an XML tag encoding its shape will be understood by default as being of the Javanese form, and will be automatically tagged in XML as having that form
- fillers for any other shape shall ideally be encoded in XML, but if necessary, you may use the shorthand §abc
 - where “abc” (any sequence of letters, followed by a space) will be converted into a symbol token in the XML tag to be added to the § character

4.2.3. Generic symbols

- in the terms of this Guide, a “**generic symbol**” is any symbol that does not meet the criteria for punctuation marks and space fillers as defined above
- note that auspicious (*maṅgala*) symbols should never be transliterated as the words *siddham* or *oṃ*
- as outlined in the introduction to this section (§4.2), generic symbols shall not be represented at the level of transliteration and should ideally always be handled in XML markup (supplemented by human-readable description)
 - the essential feature of the relevant XML markup is the use of symbol tokens (EGD §4.2.3)
- to simplify your work, especially when you are creating an e-text that will not (yet) be marked up in XML, you may choose either of the following shorthand methods for representing miscellaneous symbols
 - as tokens, using §abc
 - where “abc” (any sequence of letters, followed by a space) will be converted into a symbol token in the XML tag to be added to the § character
 - as dingbats, using any Unicode character approximating the original glyph (e.g. ♠ ♣)

4.3. Space

- spaces in your text must be encoded in XML as per EGD §4.3
- to simplify your work, especially when you are creating an e-text that will not (yet) be marked up in XML, you may use the shorthand markup _ (underscore) where an inscription employs an interword space (large enough to be called a space but smaller than the width of two average characters)
 - it is recommended that you also use a regular space before and after the underscore, but this is not required
- any other spaces — such as space left blank for filling later, or because of a defect or feature of the material — can only be handled in XML

References

- Brookes, Stewart, Peter A. Stokes, Matilda Watson, and Débora Marques de Matos. 2015. 'The DigiPal Project for European Scripts and Decorations'. In *Writing Europe, 500-1450*, edited by Aidan Conti, Orietta Da Rold, and Philip Shaw, NED-New edition, 25–58. Texts and Contexts. Boydell and Brewer.
- Coulmas, Florian. 2006. *The Blackwell Encyclopedia of Writing Systems*. 4th ed. Oxford: Blackwell.
- Damais, Louis-Charles. 1955. 'II. Etudes d'épigraphie indonésienne, IV : Discussion de la date des inscriptions'. *Bulletin de l'École française d'Extrême-Orient* 47 (1): 7–290.
<https://doi.org/10.3406/befeo.1955.5406>.
- ISO15919:2001 = International Standard ISO 15919. Information and Documentation — Transliteration of Devanagari and Related Indic Scripts into Latin Characters. Geneva: International Organization for Standardization. <https://www.iso.org/standard/28333.html>.
- ISO/IEC 10646:2017(E) = *International Standard ISO/IEC 10646. Information Technology — Universal Coded Character Set (UCS)*. 5th ed. Geneva: International Organization for Standardization.
https://standards.iso.org/ittf/PubliclyAvailableStandards/c069119_ISO_IEC_10646_2017.zip.
- Ollett, Andrew & Sarah Pierce Taylor. forthcoming. *Representing Kannada Text*. [consulted in a draft stage]
- Wellisch, Hans H. 1978. *The Conversion of Scripts—Its Nature, History, and Utilization*. New York: Wiley.