Province of British Columbia

Indigenous Languages Technology Standard

Version 1.0

February 6, 2024

Table of Contents

Purpose
The Indigenous Languages Technology Standard outlines the requirements for government IM/IT systems to be able to read, write, store, process, and display Indigenous languages.

Description
Action 3.15 of the Declaration Act Action Plan commits the Province to "adopt an inclusive digital font that allows for Indigenous languages to be included in communication, signage, services and official records." To achieve this, the Province's IM/IT systems need to meet specific technical capabilities.

This standard helps ministries ensure IM/IT systems can support Indigenous languages to deliver more inclusive government services to Indigenous people living in B.C.

By supporting Indigenous languages in systems and services, government will be able to:
- Record Indigenous-language names for people, places, or businesses.
- Display Indigenous-language names in provincial applications, data, and mapping systems.
- Print signs, correspondence, and documents that contain Indigenous languages.

This standard is to be used in conjunction with policy and resources intended to support Indigenous languages in systems and services. Additional guidance on how to support Indigenous languages in systems is available.

Application
All entities (hereafter, "ministries") identified in Core Policy and Procedures Manual Chapter 1, section 1.2.4.

Authority
Core Policy and Procedures Manual Chapter 12.

Advice on this standard
For questions or comments regarding this standard, please contact:
BC Data Service, Ministry of Citizens' Services

Review
This standard will be reviewed annually and updated on an as-needed basis.

Requirements

Supporting Indigenous languages in government systems and official records requires many elements of IM/IT systems at many levels of the systems' technology stacks. This standard outlines the technical requirements and considerations in these standard IM/IT elements:

Technology Stack Level	**IM/IT System Element**
User Interface	Unicode character set
	Character encoding
	File format
	Programming language
	System programming
	Font
	Web page
Services	API
	System programming
	Web services
	Web server
Middleware	Encryption
	Application server
Data Storage and querying	Unicode character set
	Character encoding
	Database management system
Operating systems and programming languages	Unicode character set
	Character encoding
	File format
	Programming language

For each element, found in alphabetical order below:

- the element is briefly defined,
- the purpose section explains the intent or function of the element,
- the considerations section identifies technology considerations relevant to the element.

Rather than being prescriptive, these are items that should be thought about when ensuring that an application properly deals with Indigenous language text data. Alternatives are described in the Guidance.

Ministries must test the IM/IT systems under their control to ensure they are able to read, write, store, process, and display Indigenous languages.

1. Element: API

An API, or Application Programming Interface, is a set of procedures and functions that allow computer systems to talk to each other (e.g., exchange data).

Purpose:	APIs are an efficient and secure way to share data and expose functionality between government's digital services. 'API First' is a design principle which stipulates that an API is the first interface for a given application; it is the first component to be developed over the data layer, and it is self-described.
Considerations:	1.1 Consistent metadata and encoding ensures that APIs are interoperable across organizations and helps to maintain data consistency. 1.2 Unicode Transformation Format-8 (UTF-8) is the standard encoding type for all text and textual representations of data through APIs. It must be used for all APIs published across the BC Government for both internal and external consumption.

2. Element: Application server

An Application Server is computer software that runs business applications. It is positioned as an intermediary between clients (e.g., people using web browsers) and "back-end" resources (e.g., database management systems).

Purpose:	An application server processes requests from clients through a communications channel. It provides a way to serve many clients through a centralized set of business applications.
Considerations:	2.1 The application server must be configured to support (send, receive, process, and store) Unicode data encoded with the UTF-8 (preferably) encoding or UTF-16.

3. Element: Font

The **BC Sans typeface** is an open-source font that uses the Unicode character set. It is designed to enhance the readability and presentation of digital services and communications while controlling the appearance of Unicode characters in both print and on computer screens. BC Sans is designed for and continuously improved to support Indigenous languages used in B.C.

Purpose:	Accurately displays Indigenous languages represented in government service delivery.
Considerations:	3.1 BC Sans is used in all government digital content. 3.2 All IM/IT systems use BC Sans as a display font. 3.3 BC Sans is used in all government correspondence (e.g., emails, letters, notices).

4. Element: Character encoding

Character encodings map characters to binary data (bit/byte level). Each character set uses one or more specific encodings. A system must know what encoding is used to properly interpret what is represented by the binary data. Indigenous language text consists of strings of graphemes (symbols) made up of Unicode characters. These graphemes can be encoded as binary data using the UTF-8 (preferably) or UTF-16 encoding.

UTF-8 is the preferred encoding to use on B.C. government systems, as it stores all ASCII characters using just one byte each, and most text data in B.C. government systems is ASCII. Multi-character graphemes and Indigenous language non-ASCII characters require additional bytes.

UTF-16 is also a standard for encoding Unicode characters, but it requires either 2 or 4 bytes for each character. Since UTF-8 is more efficient for applications that use mostly ASCII characters, it is the preferred encoding to use in B.C. government systems.

Purpose:	Provides a consistent way of interpreting binary data as character data.
Considerations:	4.1 All IM/IT systems use either UTF-8 (preferably) or UTF-16 to encode Unicode characters and graphemes.

5. Element: Database management system

A **database** is a digital repository that stores data and provides capabilities for managing and sharing data.

Purpose:	Stores and safeguards the information in IM/IT systems.
Considerations:	5.1 Configuration supports Unicode characters. 5.2 Character data is encoded using UTF-8. 5.3 Indigenous language text strings are treated as sequences of characters (i.e., a single character may require multiple bytes). 5.4 Extra space is allocated for string variables and database columns because some Indigenous language characters require more space when stored as UTF-8, compared to other encodings (e.g., ASCII).

6. Element: Encryption

Data Encryption refers to the process of translating plain (readable) text into (encrypted) cyphertext.

Purpose:	Encryption protects sensitive information from being disclosed to unauthorized parties. It can be applied to both data in transit between systems and data stored in a database or other file store.
Considerations:	6.1 Encryption/decryption methods support multi-byte or varying length encodings (i.e., Unicode, UTF-8) and follow the guiding principles of: • Work with bytes, not text strings, and • Do not store encrypted data in a string type.

7. Element: File format

A **file format** is the way that data is organized in a file. Following format rules ensures that application programs can process and display the data correctly.

Purpose:	Ensures that file formats used by application systems can correctly support Indigenous languages.
Considerations:	7.1 UTF-8 encoding is supported. For example: • Comma-separated value (CSV) files must include a byte order mark (BOM) at the start of the file to render properly in Excel. • Systems embed the fonts in any PDFs they generate. Note that PDF files generated from Word, Excel, or PowerPoint that use the BC Sans font will automatically embed the font.

8. Element: Programming language

System programming is the activity of developing, integrating, and modifying an IM/IT system using various programming languages. Programming languages vary in their ability to deal with multibyte, variable length encodings such as those used with Unicode. For example, reviewing application logic and programming languages when building applications can ensure a system's ability to support Indigenous languages.

Purpose:	Ensures the development of systems and applications uses logic and programming that supports Indigenous languages.
Considerations:	8.1 Programming languages used in information systems support Unicode encoded as UTF-8 or UTF-16. 8.2 Programming languages support the libraries needed to support Unicode characters that are organized into graphemes.

9. Element: System programming

System programming is the activity of developing, integrating, and modifying an IM/IT system using various programming languages. Programming languages vary in their ability to deal with multibyte, variable length character encodings such as those used with Unicode. Special libraries are sometimes needed for dealing with Unicode data and are often needed to deal with the organization of Unicode characters into graphemes. A careful review of application logic and programming languages when building applications can ensure a system's ability to support Indigenous languages.

Purpose:	Ensures the development of systems and applications uses logic and programming that supports Indigenous languages.
Considerations:	9.1 Programs enable inputs and character limits that support Indigenous languages. Logic does not restrict inputs to only Latin characters or arbitrarily short length limits. 9.2 Since the ways characters are combined to form graphemes may not be unique, programs use Unicode normalization forms to resolve ambiguities.

10. Element: Unicode character set

Unicode is an international information technology standard for storing and processing text. It includes a character set that contains the characters, technical symbols, and punctuations used in languages and writing systems worldwide, including Indigenous languages in B.C.

Purpose:	Provides a consistent way of representing text from any of the world's languages, enabling the exchange of text data between computer systems.
Considerations:	9.1 All system components (e.g., database management systems, programming languages, web servers, etc.) support Unicode. 9.2 Systems have the capability for end users to input data easily and accurately (e.g., keyboard options).

11. Element: Web page

A web page is a document, stored on, dynamically generated by, a web server. Web pages can be displayed using a web browser.

Purpose:	Provides content to application programs or people using web browsers.
Considerations:	11.1. HTML and XML web pages must define the content type character set as UTF-8. 11.2. Data in pages served must be encoded in UTF-8. 11.3. HTTPS and HTTP headers should include a charset=utf-8 in the Content-Type header element.

12. Element: Web server

A web server is a computer that processes web requests and returns web pages viewed in web browsers. Web servers can also run web services, which deliver content over the internet to application programs.

Purpose:	Provides content to people using web browsers or application programs via web services.
Considerations:	12.1. Web servers must be configured to deliver content using the Unicode character set, encoded as UTF-8.

13. Element: Web service

A web service is a computer program that runs on a web server, taking requests from, and returning results to, another computer program or user.

Purpose:	Provides content to application programs or users calling an API.
Considerations:	13.1. Web services must work with the Unicode character set and use UTF-8 (preferably) or UTF-16 when reading, writing, processing, and storing character data.

Definitions

ASCII is both a character set and an encoding. It includes lower and upper-case Latin alphabet (a-z, A-Z), numeric digits (0-9) and some punctuation characters. This very basic set is encoded as integers between 1 and 127 and each character is stored in a single byte, using 7 of the 8 bits in a byte. There are extended ASCII sets which include accented Latin characters (e.g., è). Two commonly used extended ASCII character sets are Windows-1252 and ISO-8859-1. These extended ASCII characters use all 8 bits of a byte.

Graphemes are the smallest functional unit of a writing system (e.g., ?, 7, ə, ê, or ʼ). A grapheme may be a single Unicode character or a combination of Unicode characters making up a single symbol. The symbols e and é are single-character and multi-character graphemes, respectively. Many B.C. Indigenous languages use graphemes.

Indigenous languages refers to the languages used by Indigenous peoples in B.C.

Technology Stack (Tech Stack) refers to the layered set of technologies used in building and operating computer systems. While each computer system has its own specific tech stack, the tech stack technologies can be classified into tech stack categories. This categorization is useful as it reflects the various skills and roles involved in building and maintaining systems (e.g., Front End Developers, API developers, Database Administrators, etc.)

Unicode, a character set standard maintained by the Unicode Consortium that currently defines over 140,000 characters, with the goal of encompassing characters from all living and historic language scripts worldwide. This standard is updated annually to include additional characters.

UTF-8 is a standard for encoding Unicode characters that uses 1-byte encoding for all ASCII characters, and 2, 3, or 4-byte encoding for non-ASCII characters (e.g., characters found in some Indigenous languages). Because UTF-8 is very space efficient for applications that use mostly ASCII characters, it is the preferred encoding to use in B.C. government systems.

UTF-16 is a standard for encoding Unicode characters that uses 2 or 4-byte encoding for all characters. Characters found in most writing systems (including those used for Indigenous languages) require 2 bytes. UTF-8 is the preferred encoding to use in B.C. government systems.

Supporting documents

There is additional information and resources to support adoption of the Indigenous Languages Technology Standard and to more broadly support Indigenous languages in systems: How to support Indigenous languages in systems.

Revision History

Version	Date	Notes
1.0	Feb 6, 2024	Approved by Hayden Lansdell, ADM, BC Data Service