

**Title:** Revised CJK Consolidated Requirements of Input Method

**Date Assigned:** 2005-06-22

**Source:** SWG1 meeting chairperson

**Backward Pointer:** SWG1 N001, N009

**Document Type:** Requirements document

**Status:** At the 2nd SWG1 meeting SWG1 resolved to add detailed technical requirements into SWG1 N001 as sub-subsections, without modifying the existing contents of N001.

A workshop was held at the 2nd SWG1 meeting for revising requirement document. The members of the workshop were James Su (Chair), Hideki Hiura, Roger So, Hwanjin Choe, Koushi Takahashi and Hiroshi Miura. The workshop successfully incorporated N009, Chinese requirements document, into N001 consolidated requirement document, and created this new document N011. This documents replaces SWG1 N001. It should be noted again that the existing contents of N001 were never modified and only addition of deatailed technical requirements was done.

The revision will be finalized at the 3rd SWG1 meeting in Beijing

## CJK Consolidated Requirements of Input Method

### 1. Terms and Definitions

#### 1.0 IM

Abbreviation of Input Method

#### 1.1 IM engine

An implementation of one or more input methods logics.

#### 1.2 IM logic

A logic implemented in IM engines, such as Chewing or PinYin methods implemented in a Chinese IM engine.

#### 1.3 User Interface Component

An externalized component which handles the user interface related operations on behalf of IM engine, as if it is an extended part of IM engine. User Interface Components and IM engines can be provided and used independently.

#### 1.4 IM framework

A framework which manages IM engines and User Interface Components.

#### 1.5 IM engine SPI(Service Provider Interface)

An interface among IM engines, User Interface Components and IM framework.

#### 1.6 Chewing / PinYin

Popular IM logics used in Greater China.

### 2 General requirement

The IM engine SPI standard should be used for all scripts without modification.

### 3 Functional requirement

3.1 IM engine SPI should allow application programs and IM engines to run on separate and heterogenous platforms.

3.1.1 IM engine SPI should allow conformant IM engines to be neutral to any conformant IM framework.

3.2 IM engine SPI should provide a mechanism for accommodating IM engine specific extensions.

3.3 IM engine SPI should be able to accommodate advanced input sources such as handwriting and voice as well as ordinary input sources such as keyboard and pointing device.

3.3.1 Input sources include any hardware or software input to IM, for example, keyboard press/release event, and candidates list paging/selectiing event.

3.4 IM engine SPI standard should allow subsetting of the SPI. The way of subsetting allowed should be clearly defined in the standard.

3.5 IM engine SPI should allow multiple programming language bindings.

3.5.1 The reference IM engine SPI should be in ISO C99.

3.5.2 Other language bindings are out of scope of the IM engine SPI specification.

3.6 IM engine SPI should be language agnostic and should allow the use of multiple languages, multiple IM engines per language, multiple languages per IM engine, multiple IM logics per IM engine and allow a user to dynamically switch among them at runtime.

3.6.1 IM engine SPI should not depend on system language setting.

3.7 IM engine SPI should allow IM engines to implement their own licensing schemes.

3.8 IM engine SPI should allow character data stream/string to be at least in the Unicode coded character set including Variation Selectors.

3.9 IM engine SPI should allow the control of User Interface Components from IM engine side such as panel, status window and palette.

3.9.1 'Control' includes bi-directional communication.

3.10 IM engine SPI should provide platform-independent, secure, multi-user handling capability, including data stream management, and storing and retrieving per-user data of reasonable size, such as configuration, dictionary and key bindings.

3.11 IM engine SPI should allow various input context handling schemes.

3.12 IM engine SPI should allow IM engine to access and modify data in application domain.

3.13 The IM engine SPI standard should be specified in multiple parts, a generic interface and architecture specific interfaces.

3.14 IM engine SPI should specify User Interface Components to be externalized from IM engine and also specify the method to interact between them. IM engine SPI should include the interface between IM framework and User Interface Components which may be in a different SPI from the interface between IM framework and IM engine.

3.15 IM engine SPI should not preclude non Open Source implementations.

3.16 IM engine SPI should include a method to accommodate IM engines' specific key-binding requirements.

Uncategorized

2.5) Functionality for IM engine and User Interface Component to commit string and keyboard event to client application.

3. All memory allocate/free operations required by IM engine SPI should obey the rule: Who allocate the memory should be in charge of freeing it. In short "Who allocate, who free".

===== end of document =====