WHAT IS INTERNATIONALIZATION (I18N), LOCALIZATION (L10N) AND GLOBALIZATION (G11N)

In computing world, internationalization (i18n) and localization (L10n) are two popular terms that refer to adapting software application to different languages, regional differences and technical requirements of a target market. This concept is also known as NLS (National Language Support or Native Language Support).

Internationalization (i18n) is the process of designing a software application so that it can be adapted to various languages and regions — currency, number separators, dates and so on — without the need for redesign.

In i18n, the common abbreviation for internationalization, the 18 stands for the number of letters between the first i and last n in internationalization, a usage coined at DEC in the 1970s or 80s.

Localization (L10n) is the process of adapting a product or software to a specific language or culture so that it seems natural to that particular region. True localization considers language, culture, customs and the characteristics of the target locale.

It frequently involves changes to the software’s writing system and may change keyboard use and fonts as well as date, time and monetary
formats. In L10n, the common abbreviation for localization, the 10 refers to the ten letters between the l and the n. The capital L in L10n helps to distinguish it from the lowercase i in i18n.

Globalization (g11n) refers to the process that addresses business issues associated with launching a product globally, such as integrating localization throughout a company after proper internationalization and product design. In g11n, the common abbreviation for globalization, the 11 refers to the 11 letters between the g and the n.

Some companies, like IBM and Sun Microsystems, use the term Globalization for the combination of internationalization and localization.

Microsoft defines Internationalization as a combination of World-Readiness and localization. World-Readiness is a developer task, which enables a product to be used with multiple scripts and cultures (globalization) and separating user interface resources in a localizable format (localizability).

Microsoft World-readiness: This facet covers generic coding and design issues and comprises two major areas – globalization and localizability.

Microsoft Localization: This involves translating and customizing a product for a specific market.

For example, using the National Language Support (NLS) supplied by the Microsoft Win32 application programming interface (API) is a world-readiness step, whereas modifying the user interface (UI) elements, translating text, and standardizing terminology are localization steps.

Developers write code, so they tend to focus primarily on world-readiness issues. But because code and feature design affect how a product is
translated and customized, developers must also understand basic localization concepts.

Microsoft Globalization: The process of developing a program core whose features and code design are not solely based on a single language or locale. Instead, their design is developed for the input, display, and output of a defined set of Unicode-supported language scripts and data related to specific locales.

Microsoft Localizability: The design of the software code base and resources such that a program can be localized into different language editions without any changes to the source code.