-- --

IEEE P1003.0 Draft 14 – November 1991

5.1 Internationalization

Responsibility: Ralph Barker

Editor's Note: Almost all instances of ''must'' in this clause have been changed to ''should'' without further diff marks. E E

5.1.1 Overview and Rationale

Historically, information systems intended for use within a particular national or cultural market have been designed specifically for the requirements of that market. If the vendor or developer was based in a country other than that of the target market, this was typically accomplished through substantial re-engineering the features of an existing system designed for some other country, and doing so at considerable cost. As the developer desired to market the system in additional countries, the process of re-engineering was repeated for each new national or cultural market. Application software developers were faced with the same problem. The very nature of this style of development produced little concern for portability across national or cultural boundaries, or interoperability between them. Users or organizations that needed to operate in multiple national or cultural markets typically did so with multiple, generally incompatible, information processing systems.

The interfaces provided by the POSIX Open System Environment (POSIX OSE) can be generalized, however, through the use of internationalization, to extend across national and cultural boundaries. Such a model provides the foundation for international portability of application software, increased user portability, and enhanced interoperability and data exchange capabilities. The task of internationalization is to ensure that the services provided by the POSIX OSE, and the interfaces between such services, are specified in such a way that they can be easily used all over the world. Additionally, as the user is likely to require services from any or all of the service categories of the POSIX OSE, internationalization impacts all areas of the POSIX OSE, and should be viewed as an OSE Cross-Category Service. Since the internationalization aspects of general OSE services and application program interface (API) services are similar for all of the POSIX OSE service categories, they are discussed here rather than repeating them in each of the services sections within this guide.

The ability of the service categories of the POSIX OSE to support multiple natural languages, and the underlying cultural conventions, is a two step process. These two steps are generally referred to as ''internationalization'' and ''localization.'' First, the interfaces between the service categories are generalized, so that they are not oriented to the requirements of any particular natural language or set of cultural conventions (internationalization). Then, facilities are provided by the POSIX OSE that allow the user to select the desired natural language and cultural conventions (localization). Tools are provided to facilitate this process.

Within this context, cultural conventions, while discussed more fully later in this clause, may be viewed as various aspects of how information is presented to the user. Different cultures, for example, use different formats for dates and numeric values and use different currency symbols. The interfaces provided by the POSIX OSE should allow the information to be presented to the user in the appropriate format as well as the appropriate natural language.

5.1.2 Scope

The POSIX OSE provides services that are necessary to support users, irrespective of their particular natural language or cultural conventions. While it is not expected that every implementation of the POSIX OSE would provide support for all possible natural languages and cultural conventions, the specification of the services and the interfaces within the POSIX OSE should not preclude such support. In addition to the service and interface requirements described here, it should be noted that internationalization is affected by a number of elements that are beyond the scope of this guide. Actual implementations of the internationalized POSIX OSE, for example, may need to consider the impact of multiple sets of governmental and regulatory agencies, international data communication standards and other elements which are presently not specified within the POSIX OSE, such as data portability between localized information processing systems.

Service requirements differ from country to country and even between users within one country. Many users, for example, may require the simultaneous support of multiple natural languages and cultural convention sets. Therefore, the basic internationalization requirement within the POSIX OSE is to provide a set of services and interfaces that allow the user to define, select, and change between different culturally related application operating environments supported by the particular implementation. Specifically:

— The POSIX OSE should provide the means of adjusting the output of specific functions and utilities to support different natural languages, cultural conventions and character sets as may be required by the supported natural languages.

— A user should have the capability to select an internationalized user environment that specifies a particular set of data presentation characteristics, including cultural conventions, character sets and native language.

— An implementation of the POSIX OSE should be able to concurrently support different applications functioning in different internationalized user environments, supplying different sets of natural languages, cultural conventions and character sets for different users.

— The capability of supporting different internationalized user environments, and the associated natural languages, cultural conventions and character sets, should not require any changes to the logic of existing application programs.

— The effect of the user selecting a new internationalized user environment, and its associated natural language, cultural conventions and character set, should be transparent to application programs.

— The model should be flexible, to support future extensions and requirements.

5.1.3 Reference Model

Internationalization is an OSE Cross-Category Service, spanning all OSE service categories. While various reference models have been used in published technical papers to depict internationalization issues, the internationalization services described in this clause conform to the POSIX OSE Reference Model.

5.1.4 Service Requirements

The POSIX OSE should provide services on different levels: general service requirements to be satisfied for any requesting program; API service requirements to be satisfied at the application program interface for a specific program; and a set of tools to support the localization of systems and applications. This subclause (5.1.4) will discuss these different service requirements in detail. In examining these service requirements, it is helpful to draw a distinction between those services which are required to support the portability of an application platform across cultural boundaries, and those services which are required to support the portability of an application across one or more sets of cultural conventions which may be supported on a single application platform.

5.1.4.1 General Service Requirements, Application Platform

Internationalization requirements are focused on support and handling of:

— Character sets and data representation

— Cultural conventions

— Natural language support

5.1.4.1.1 Character Sets and Data Representation

The character set for the English language can easily be satisfied by the standard ASCII character set (American Standard Code for Information Interchange). The ASCII code uses 7 bits to uniquely identify each of the 95 available characters. For European and American languages beside English, the number of local characters is much larger. The far-east requirements for thousands of pictograms add yet another dimension to the coding rules and techniques.

Different standards address the methods by which the local character repertoires can be coded for unique identification. While replacement of seldom-used characters in the 7-bit codings can support a single additional language besides English, 8-bit coding schemes are used to satisfy multiple languages concurrently by assigning an additional 96 graphic characters to the available repertoire. An example is ISO 8859-1 (the extended ASCII code), which can support all of western Europe, America, Australia, and other English speaking countries all over the world. For Eastern Europe, Greece, Russia, Arabia, and many other countries, other 8-bit codes are defined. Japan, China, Korea, and Taiwan have so many characters in their repertoire that 16 bits are needed to identify them clearly. Work is under way to develop a multi-octet character set with up to 32 bits per coded character; this method will allow concurrent use of all possible languages in the same application.

Because different coding schemes are used, it is important that the application platform have the potential capability of supporting all of them. It is also important that the application platform has the capability to represent (display, print) the data correctly. It is also important that an application be able to determine in which coded character set data items are stored on disk or tape. Otherwise, it is impossible for the application to interpret the data correctly. Currently the user must control the consistent use of the same coded character set within an application, but in the future the application platform should be able to provide identification methods for the coded character sets used for data storage, processing, communication, and presentation. It might also be advantageous for the application to be able to prohibit users from updating data stored in one coded character set with data in another coded character set since this would immediately corrupt data bases or flat files. Therefore it may be necessary in the future to provide a method of announcing the coded character set in which data are stored, processed, communicated, and presented.

The general service for support of character sets and data representation in an international environment are:

(1) Coded character set independence: the ability of the application platform to input, store, manipulate, retrieve, communicate, and present data independent from the coding scheme used. This includes 7-bit, 8-bit, 16-bit, and multi-octet coded character sets.

(2) Character set repository: the ability of the application platform to maintain and access a central character set repository. This repository contains all coded character sets used throughout the platform and specifies relevant information about them:

- Code format: the repository contains information, if characters are coded in 7 bits, 8 bits, 16 bits, or any other format.

- Data class definition: the definition that a character is considered numeric, alpha, etc., by the programming languages. This classification can vary for the same character from country to country.

- Collating rules: different character sets have different coding for characters. Thus, comparison of strings of such coded characters should follow rules defined for the specific character set. Culturally dependent additional collating rules are discussed in 5.1.4.1.2.

- Lower- to uppercase mapping: this defines the rules of mapping, if for a specific character no upper- or lowercase is available. Examples are the lower case umlauts which do not have uppercase representations in Switzerland; the uppercase forms are A, O, or U, respectively, followed by a lowercase ''e''.

— Escapement rules: some languages like Hebrew and Arabic are written from right to left; numbers within text in these languages are written from left to right. It is necessary to store these escapement rules with the character set.

— Presentation rules: the application platform should have the ability of providing fallback presentation rules for the presentation of coded characters that have no associated graphic shape.

(3) Character set identifier: the application platform should provide the ability to uniquely identify each coded character set to allow compatibility checks and translation or transliteration to and from other registered character sets. This ensures data integrity in the communication of data across computers and networks.

(4) Character set selection: the application platform should allow the end-user or the application to select the coded character set to be used; otherwise, the application should automatically select a default coded character set according to preset parameters. It should be possible to switch to other coded character sets and to invoke translation routines where required.

(5) Data announcement: the application platform could benefit from having the ability to recognize the coded character set of data entities (files, messages, etc.). One way of doing this is to store the character set identifier together with the data; standardization efforts are under way to formalize this process, with consideration being given to the level of granularity of such identification (e.g. file, word, character). The announcement enables the application to prohibit updates with data coded in other character sets, thus ensuring data integrity even in distributed systems.

(6) Data presentation: the application platform should be able to present data on different display or output devices, potentially according to rules in a repository, including escapement of characters and selection of different shapes. Preparing data for presentation may involve extensive translation and transliteration due to potential hardware limitations of the printers and displays used in a particular installation.

(7) Data communication: the application platform should be able to transmit and receive data from communication systems and to maintain the integrity of the information. In an internationalized environment, this capability might include data translation due to different coded character sets being used by different service categories of the application platform.

(8) Data input: the ability to enter data is not necessarily controlled by the application platform. The complexity of the input of Asian languages though might strongly support the idea of a standardized input mechanism interface. Depending on how other internationalization service requirements are met, it might also be beneficial for input data to carry some form of character set identification.

5.1.4.1.2 Cultural Conventions

Besides using different characters and different languages, countries throughout the world have also developed quite different cultural conventions. Even within one country we can find significantly different cultural environments. The prime example is Switzerland, where French, German, Italian, and Rhaeto Romanic are officially accepted languages. Combined with the language preferences are conventions about the formats of time, date, numeric values, and measuring systems. Currency symbols, paper formats, hyphenation, and collating are dependent on cultural conventions. End-user-oriented applications have to address these issues to provide a familiar local view, which helps to prevent operating errors.

The general service requirements for cultural conventions are:

(1) Cultural convention repository: The application platform should have the ability to store and access rules and conventions for cultural entities. These might be areas with a common language, geographic areas, or areas with common cultural or historic background. The repository should contain specifications and presentation rules for:

- Date and time formats: indicating the formats associated with the particular cultural entity. For example, while in the US the date is expressed in the format month/day/year, the European preferred format is year-month-day for data processing purposes and day-month-year in personal use. Japan counts the years according to the reign of the current emperor. Additionally, twenty-four-hour clocks, which are prevalent in Europe, are commonly used only in military circles in the US, while the terms ''am''and ''pm'',denoting morning and afternoon, are used by the general public. These are only a few examples for the cultural differences in this area. The application platform should be able to store the preferred forms for date and time for a specific cultural entity and make it available upon request in this format.

- Week and day numbering: in Europe, the week starts on Monday, in the US on Sunday. The application platform should be able to supply the requesting program with the needed information, potentially from a repository according to specified rules.

- Formats of numeric fields: handling of numeric fields in unfamiliar formats is one of the major reasons for human errors. The application platform should provide the service to format the values according to specifications in the repository. The characters that signify the decimal point (comma, period, etc.) should be defined, as well as the number of decimals, the grouping of digits before the decimal point and the presentation of negative values.

- Currency symbols and field length: the handling of currency symbols in the different cultural areas should be provided by the general internationalization services. The currency symbols might be more than one digit long and can appear before or after the currency field. The format of currency fields might differ from that of numeric fields; for example, in Portugal the $-sign is used as the decimal point. Information about these conventions should be stored in the repository and be used by the application platform for local formatting of currency fields. Not necessarily a service, but similarly important, is the understanding, that due to the value of different currencies, the field lengths should be considered carefully. Also some currencies do not have decimals (e.g., Italian Lira).

— Paper formats: internationally usable and portable applications should be able to print on different paper formats. While quart format is predominant in the US and the far east, the DIN standardized A-formats are used in Europe. Printer drivers should be able to adjust their output to local formats, defined in the cultural convention repository.

(2) Cultural repository selection: these repositories should be available to all applications. Users and applications should be able to select a repository from the application platform; a default value should be provided if no selection is made. An additional service allows dynamic switching to other repositories upon user or program requests.

(3) Collating rules: besides the generic binary and character-set-dependent sorting rules, the application platform should have the ability to sort data according to local rules, defined in the repository. An example for culture-dependent collating rules is the handling of umlauts; while they are sorted with the base characters in Austria, they are sorted at the end of the alphabet in Sweden. Adding complexity, they can be sorted differently within one country between normal business use, such as dictionaries, and in telephone books. Other idiosyncrasies are the sorting of one character as two (the German ‘‘sharp-s’’ sorts as ‘‘sz’’in Austria and ‘‘ss’’in Germany), or two characters as one (the Spanish ‘‘ch’’sorts as one character), or the position of accented characters in a string, and more. User-defined collating tables in the cultural convention repository allow culture or application-dependent sorting services.

5.1.4.1.3 Natural Language Support

The POSIX OSE should give users the ability to select a natural language for their dialogue with the system and applications. While it is unrealistic to expect all application platforms to support all possible natural languages, error messages, online documentation and help facilities, selection menus, and the relevant user interaction with these services should be prepared for translation into the supported user-selectable natural language. Additionally, the POSIX OSE should support differences between the natural language selected by the user for interaction with the application platform and that selected for use within a particular application. For word- and text-processing, the service includes hyphenation and spell checking with possible thesaurus support in different languages. The problem is complicated by the fact that data can contain text in different languages in the same document.

The service requirements for natural language support are:

(1) Multilingual capability: the application platform should be able to support more than one language simultaneously. For example, one process might be providing French language capabilities while another process operated in Japanese. The application platform should be able to let users select their preferred languages for communication with the application and allow them to switch dynamically to another language. The application platform also should have the capability to assign a default language, based on parameters for the application platform, the specific workstation, the user identification, or the application.

(2) Natural language message system: the application platform should have the capability to present (display, print, . . .) messages, menus, forms, and online documentation in the language, selected by the user. The application platform should be able to support multiple languages simultaneously for different users and it should allow the user to switch from one language to another. The following problems also should be handled correctly:

— The program code of the application should be able to be independent from any particular natural language, presenting messages in the natural language used within the internationalized user environment selected by the user.

— Variable message length: the application platform should support the presentation of messages of variable length, as translation into other languages changes the length of the message; English text is usually quite short compared to the same text in, e.g., German or Finnish. Ample room should be available in the display field to accommodate this variation.

— Inserted parameters and word order: the application platform should have the capability of inserting variable parameters into messages at the location appropriate for the user selected natural language.

(3) Support of local keyboards: the application platform should be able to correctly interpret the input from keyboards that have been modified locally to support the local character sets.

(4) Local language user interaction: the application should be able to accept solicited input from the user in the language selected by the user, without dependence within the application logic on a particular natural language or set of cultural conventions. For example, many applications use the first characters of prompts to make selections; this method is not acceptable in an internationalized system. The translation process changes the prompts and with them their first character; more than one prompt could have the same start-character and the program logic would not work. Multiple languages should be supported simultaneously.

5.1.4.2 API Service Requirements

All the general services defined in 5.1.4.1 should be accessible from the applications through requests to the application program interface. The API service requirements can be structured in the same way as the general requirements, which they call for.

5.1.4.2.1 Cultural Conventions

— Cultural convention invocation: the application platform should allow the application to invoke a specific cultural convention from the repository. It should automatically invoke the default convention set, if no selection is made by the application.

— Cultural convention change: when requested by the application or the user, the application platform should change the used cultural convention dynamically.

— Provide local values: upon request from the application, the application platform should return local formats for time, date, calendar, numeric fields, currency fields and symbols.

— Local sort and comparison: when requested by the application, the application platform should compare and sort data according to the local collating rules defined in the cultural convention repository.

5.1.4.2.2 Natural Language Support

— Language selection: the application platform should present messages, menus, forms, online documentation, and user interaction in the natural language selected by the user or automatically by the system based on preset parameters for the application, the session, the user, or the system.

— Change of language: upon request from the user, the application platform should be able to dynamically change, prior to the invocation of a particular user application, the language used for messages, menus, forms, online documentation, and user interactions.

5.1.4.3 Localization Tools Requirements

Internationalization of application platforms and applications is the basis for their localization in the different countries. It is important for the user that this localization can be performed in a well prepared, organized way without the need to know the internal structure of the application platform or the application. The following requirements for localization tools are key to successful localization of application platforms and applications:

— Character set repository tools: tools should be provided to set up and maintain character set repositories. They also should allow the addition of new character sets to the repository.

— Cultural convention repository tools: tools should be provided to set up and maintain the cultural convention repositories. Addition of new cultural environments should be possible. User-definable collation tables are essential parts of these repositories; tools to define and maintain them should be offered.

— Translation support tools: facilities for the set-up and maintenance of local language message files, menus, forms, online documentation, and user interaction tables should be provided. The addition of new supported languages should be allowed by such tools. Additionally, any such translation tools should allow revision control, so that only new or changed text would require translation for new software releases.

5.1.5 Standards, Specifications, and Gaps

There are not many standards available that deal with internationalization. The majority of current standards describe character sets, both for control characters and for graphic characters in different coding schemes (7-bit, 8-bit, etc.). A few standards address the formats of time and date, and some standards touch peripherally on the subject of data announcement.

An example of how cultural conventions and languages are currently supported is the *locale* () function. It allows the application developer to select portions or all of predefined support features for national languages and local cultural conventions. The portions, called categories, correspond to the areas of functionality; presently supported are character classification, collation sequence, date/time format, monetary format, and numeric format. Other categories, such as message handling, are likely to be implemented, too. Other systems have started to implement similar philosophies of general services to support local cultural conventions.

5.1.5.1 Current Standards

5.1.5.1.1 International Standards

— ISO 646: 1983, *ISO 7-Bit Coded Character Set for Information Interchange*

Defines the binary representation of 128 control, (Latin) alphabet, digit, and symbol characters. Describes in general the use of the control characters. Describes option of national replacement characters.

— ISO 2014: 1976, *Writing of Calendar Dates in All-numeric Form*

This international standard specifies the writing of dates of the Gregorian calendar in all-numeric form, signified by the elements year, month, and day.

— ISO 2022: 1986, *ISO 7-Bit and 8-Bit Coded Character Sets—Code Extension Techniques*

Table 5-1 – Internationalization Standards E

Service	Type	Specification	Subclause	
Character set/data representation	S	ISO 646, ISO 2022, ISO 4031, ISO 4217, ISO 4873, ISO 6093, ISO 6429, ISO 6936, ISO 6937-1, ISO 6937-2, ISO 7350, ISO 8601, ISO 8859-*n* (1-9), CCITT T.61, GB 2312, JIS X 0208, KS C 5601	5.1.5.1	E
Character set/data representation	E	ISO DIS 10367, ISO DIS 10646	5.1.5.2	E
Cultural convention	S	ISO 2014, ISO 3307	5.1.5.1	E
Natural language support	E	ISO/IEC 9995-x, CSA-Z243.200-88	5.1.5.2	E

Defines techniques for expanding the number of characters represented by the base character set.

— ISO 3307: 1975, *Representation of Time of the Day*

This international standard is designed to establish uniform time representation based upon the 24-hour timekeeping system. It provides a means for representing local time of the day and Universal Time in digital form for the purpose of interchanging information among data systems.

— ISO 4031: 1987, *Representation of Local Time Differentials*

This international standard specifies a standard means for representing local time differentials to facilitate interchange of data among data systems.

— ISO 4217: 1987, *Codes for the Representation of Currencies and Funds*

Specifies the representation of currencies and currency symbols

— ISO 4873: 1986, *ISO 8-Bit Code for Information Interchange—Structure and Rules for Implementation*

Outlines the structure of the ISO 8-bit code and rules for implementation.

— ISO 6093: 1985, *Presentation of Numerical Values in Character Strings for Information Interchange*

Specifies three presentations of numerical values, which are represented in character strings in a form readable by machine, for use in interchange between data processing systems. Also provides guidance for developers of programming languages standards and Implementor's of programming products. These representations are recognizable by humans, and thus may be useful in communication between humans.

— ISO 6429: 1988, *ISO 7-Bit and 8-Bit Coded Character Sets—Control Functions for Coded Character Sets*

Defines control functions and their coded representations for use in a 7-bit code, an extended 7-bit code, an 8-bit code, or an extended 8-bit code. Specifies a C0 set, a C1 set, control functions derived there from, and a number of independent control functions.

— ISO 6936: 1988, *Conversion between the Two Coded Character Sets of ISO 646 and ISO 6937-2 and the CCITT International Telegraph Alphabet No. (ITA) 2*

Specifies the rules for conversion between ITA 2 representation of 58 characters and the ISO 646 representation of 128 characters.

— ISO 6937-1: 1983, *Coded Character Sets for Text Communication—Part 1: General Introduction*

Defines terms and concepts used in describing and using code representations of character sets.

— ISO 6937-2: 1983, *Coded Character Sets for Text Communication—Part 2: Latin Alphabetic and Non-alphabetic Graphic Characters*

Defines a repertoire of Latin alphabetic and non-alphabetic characters. Specifies binary representation of the characters. Specifies rules for the definition and use of character sets that are subsets of the repertoire.

— ISO 7350: 1984, *Registration of Graphic Character Subrepertoires*

Specifies the procedures for preparing, registering, publishing, and maintaining the register of graphic character sets that are composed from the character repertoire of ISO 6937 and the procedures for assigning identifiers to the sets.

— ISO 8601: 1988, *Representation of Dates and Times*

Specifies the representation of dates A.D. in the Gregorian calendar and times and representation of periods of times. Applicable whenever dates and times are included in information interchange.

— ISO 8859-x: 1987, *8-Bit Single-Byte Coded Graphic Character Sets*

Specifies a set of up to 191 graphic characters by means of a single 8- bit byte. The versions (‘‘-x’’) indicate different coded character sets:

-1 Latin Alphabet No. 1

-2 Latin Alphabet No. 2

-3 Latin Alphabet No. 3

-4 Latin Alphabet No. 4

-5 Latin/Cyrillic Alphabet

-6 Latin/Arabic Alphabet

-7 Latin Greek Alphabet

-8 Latin/Hebrew Alphabet

-9 Latin Alphabet No. 5 E

— CCITT T.61, 1985: *Character Repertoire and Coded Character Sets for the International Teletex Service*

Describes detailed definitions of the repertoires of graphic characters and control functions to be used in the international Teletex service. The means by which supplementary character repertoires are defined are also described.

5.1.5.1.2 Regional Standards

Presently, no regional internationalization standards which relate to the scope of this guide have been adopted.

5.1.5.1.3 National Standards

Many of the international ISO standards have ''twins'' in the national standards bodies; i.e., the same text is given a local standard identification. Also, national standards bodies have often developed standards for local representation of time, date, and currency. The implementation of these standards into an internationalized system is a prime example of localization.

Here are some standards that have no international equivalent:

— GB 2312: 1980, Chinese national character set standard

— JIS X 0208: 1983, Japanese national character set standard

— KS C 5601: 1987, Korean national character set standard

5.1.5.2 Emerging Standards

5.1.5.2.1 International Standards

The rapid development of business opportunities in the Pan-European and the Asian market has spawned a wealth of activities to develop standards for the support of internationalization in the field of information technology. These emerging standards deal with character sets, language neutral user interfaces, and communication.

— ISO DIS 10646: *Multiple Octet Coded Character Set*

This standard will permit the presentation of all of the world's scripts in computer based systems, and their unambiguous interchange between one system or person and another. It is applicable to the representation, processing, storage and presentation of the written form of the languages of the world.

— ISO/IEC DIS 10367: *Repertoire of Standardized Coded Graphic Character Sets for Use in 8-Bit Codes*

This standard specifies a unique graphic character set for use as G0 set and a series of coded graphic character sets of up to 96 characters for use as the G1, G2, and G3 sets in versions of ISO 4873. All sets specified in this standard are shown as elements of an 8-bit code.

— ISO/IEC CD 9995-x: *Information Technology—Keyboard Layouts for Text and Office Systems*

This family of standards defines the layout of keyboards so that they can be used for input of multilingual information.

5.1.5.2.2 Regional Standards

The European Community is in the process to define European standards, called EN (Europaeische Norm). No internationalization standards have yet been adopted.

5.1.5.2.3 National Standards

National standards under development which relate to internationalization include:

— CSA-Z243.200-88: *Canadian National Keyboard Standard for the English and French Languages in Text and Office Systems*

5.1.5.3 Gaps in Available Standards

5.1.5.3.1 Public Specifications

The PC character set was defined at a time, when the international standards for single-byte, 8-bit character sets were not available yet. Therefore, the PC character set was accepted and still is a de facto standard in the PC world. The concept of different code pages has been implemented in MS-DOS and WINDOWS-3 is using ISO 8859-1 internally for compatibility reasons with other systems. Some companies have gone similar routes and developed their own, multilingual character sets for specific applications, the general trend is clearly towards ISO standards wherever they exist.

A consortium of software and hardware companies is developing ‘‘Unicode,’’ a 16-bit character set standard for broad international use. E

5.1.5.3.2 Unsatisfied Service Requirements E

While the character set arena is heavily populated, very little work is done in other areas of internationalization of products. Standards should be developed for:

— Cultural conventions repository

— Application program interface services for cultural conventions

— Application program interface services for character set handling

— Multilingual collating rules

— Input methods interface for Asian languages

— Standards for message delivery systems

— Data announcement standards

Additionally, no standards currently exist that support the following character set and data representation functionality:

(1) Character set invocation: the application platform should allow the application to invoke a specific character set from the character set repository. It should automatically invoke the default character set, if no selection is made by the application.

(2) Character set changes: When requested by the application, the character set should be changed dynamically.

(3) Character set identifier: the application program should be able to write the character set identifier to data and should be able to retrieve the identifier for requested data.

(4) Character set identifier comparison: the application platform should, upon request from the application or automatically, compare the character set identifiers of interacting data in the application (input, processing, data storage, communication, and output).

(5) Character set translation: the application platform should provide translation of character sets, when requested by the application or automatically, when detecting a mismatch in the comparison process.

5.1.6 OSE Cross-Category Services

Not applicable.

5.1.7 Related Standards

The nature of internationalization as being a cross-component facility is that it affects just about every element in the information processing world. Thus, almost all standards in this environment are related to the subject. Here we will point out a few major families of standards, strongly related to internationalization.

— ISO DIS 8613: Office Document Architecture and Interchange Format (ODA)

This family of standards, ODA/ODIF, consist of:

1.2 Introduction and General Principles

2.2 Document Structures

3 Document Processing Reference Model

4.2 Document Profile

5.2 Office Document Interchange Format

6.2 Character Content Architectures

7 Raster Graphics Content Architectures

8 Geometric Graphics Content Architectures

— ISO 8824: 1987, *Specification of Abstract Syntax Notation One ASN. 1*

Specifies a notation for the definition of abstract syntaxes, enabling Application Layer standards to define the types of information they need to transfer using the Presentation service. It also specifies a notation for the specification of values of a defined type.

— ISO 8825: 1987, *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN. 1)*

Defines a set of encoding rules that can be applied to values of types defined using the notation specified in ASN. 1. Application of these encoding rules produce a transfer syntax for such values. It is implicit in the specification of these encoding rules that they are also be used for decoding.

— All programming language standards, since programming languages have to support internationalization, and have to work correctly in localized environments. Their generated code itself has to work ‘‘localized.’’

E

5.2 System Security Services

Responsibility: Michelle Aden

5.2.1 Overview and Rationale

Information is the key to successful use of a system. For example, if used effectively and efficiently, information may be used to underpin enhanced service and to aid the derivation of strategic plans. Much of this information, for example, personal customer details and business financial plans, will be of a sensitive nature.

Although authorized users may be able to take advantage of the POSIX Open System Environment (OSE) to increase productivity and efficiency, unauthorized individuals may also be able to take advantage of the OSE to steal, manipulate or to deny others access to information held within the system, or to deny involvement in some transaction performed via the system.

Security services must therefore be provided within the system if it is to prevent these unauthorized activities. To achieve an optimum degree of confidence in the correctness and effectiveness of a system's security services, a system specific security policy must be derived and appropriate security functionality designed into the system at the beginning of its life cycle.

A relatively high degree of protection for ordinary computer systems can be achieved if system administrators correctly configure and maintain the system according to recommended security guidelines and practice, such as those described within the *X/Open Security Guide* . However, additional security facilities must be supported within the system to achieve protection against the small percentage of attackers who are noncasual, and who are determined to breach the security of the system. It is the intent of the security extensions to the base POSIX interface standard to support these additional security facilities.

The four basic security objectives of a system are to maintain:

— Confidentiality. The system must prevent unauthorized viewing of data.

— Integrity. The system must prevent unauthorized alteration or deletion of data.

— Availability. The system must ensure that authorized users are not prevented from accessing and processing data.

— Accountability. The system must ensure that users are made accountable for their actions, for example to ensure that users are correctly billed for system usage. See also 5.3.4.11. E

Different user groups may place different emphases upon these four basic security objectives. For example, the military security sector may place more importance upon confidentiality than accountability while, correspondingly, the commercial sector may place more importance upon accountability than confidentiality.

5.2.2 Scope

One of the goals of system security is to provide defense in depth, such that if one layer of security is breached then further layers of security will limit and/or prevent unauthorized activities within the system.

To achieve a high degree of confidence in the correctness and effectiveness of the security of a system that will be processing sensitive information, security must be designed into the system at the beginning of its life cycle.

A System Security Policy (SSP) defines what it means for a specific system to be ‘‘secure’’ and, as such, forms the basic security input into the system lifecycle. Specification of an SSP is therefore axiomatic to the design of a secure system.

Although the SSP defines what security measures will be provided within the system, it is the system design documentation that defines how these security measures will actually be implemented.

One aspect of an SSP may be that it mandates conformance with the POSIX security extensions.

Security interface specifications are intended to assist in the construction of a secure system. They do not, in isolation, provide any protection against threats to a system.

5.2.3 Reference Model

The reference model for security is the same as the model shown in Figure 3-3. Security has an impact on all of the APIs and EEIs in the model.

5.2.4 Service Requirements

Through an analysis of the potential threats and requirements of the system, the system security objectives and hence the necessary System Security Policy (SSP) rules may be derived. This analysis must also take into account appropriate corporate, legal, and standardization requirements.

System confidentiality, integrity, availability, and accountability may be supported by the following security objectives:

Technical Security Objectives

— Identification and Authentication. A system entity, such as a user or system element, must prove that its claimed identity is legitimate, such that another system entity may place confidence in that claimed identity.

— Access Control. Access to system resources will be restricted to authorized entities only. Residual data contained within an object will be securely erased before it may be reused by a system entity.

— Accountability and Audit. System users must be made accountable for their actions. Audit trails of these actions will then be maintained and utilized such that unauthorized system activity will be detected.

- Accuracy. The system must ensure that the correctness and consistency of security-relevant information is maintained.
- Availability. System resources will be provided to users in a consistent and reliable manner.
- Data Exchange. Data transmitted between system users and/or elements will be protected from unauthorized interference or viewing. Originators and recipients of data will be authenticated and will be able to mutually prove their respective participation in the transaction.

Nontechnical Security Objectives

- Assurance. The security of the system must be specified, designed, implemented, tested, and maintained in such a way that confidence can be placed in the correct and effective operation of the system. Also, procedures must be specified to ensure continued confidence in the security of the system in the event that the system is modified in some manner.
- Security Roles and Responsibilities. Security activities must be partitioned and allocated to identifiable security administrators who will then be responsible for ensuring that their allocated task is satisfactorily performed.
- Secure Operating Procedures. Procedures must be written that will guide system administrators and users as to the correct procedure to follow in the event of some security-relevant occurrence.

5.2.4.1 Application Programming Interface Services

E

The POSIX security interfaces will support Audit, Privilege, Discretionary Access
Control (DAC), Mandatory Access Control (MAC), and Information Labels (ILs). E

The audit services include: E

- Ability to record the user identification for actions within an audit trail E
- Ability to process the audit trail E
- Ability to use the audit trail to generate alarms E

The privilege control services include: E

- Ability to grant users only the minimal security required to perform a task E

This will minimize the impact of a subverted security administrator or unauthor- E
ized usage of a security administrator role. E

The discretionary access controls (DAC) provide the following services: E

— Ability to control fine-grained user access to objects

— Ability to provide extended user access bits beyond the traditional user-group-other

— Ability to support access control lists (ACL)

The mandatory access controls (MAC) and information labels (IL) support policies for labeling:

— Ability to associate a MAC label with an object

— Ability to label information (e.g., physical document handling restrictions)

5.2.4.2 External Environment Interface Services

Note to reviewers: This subclause will be provided in a later draft. Mock ballot reviewers are welcome to submit comments on the types of services required at the EEI.

5.2.5 Standards, Specifications, and Gaps

Table 5-2 lists the current, emerging, and gaps in security standards.

Table 5-2 – Security Standards

Service	Type	Specification	Subclause
System Security	E	IEEE P1003.6 API	5.2.5.2
Access Control	E	ISO/IEC 8613	5.2.5.2
Directory Authorization	S	CCITT X.509	5.2.5.1
Security	G	ECMA CMA 138	5.2.5.3
Trusted Systems	G	DOD 5200.28- STD	5.2.5.3

5.2.5.1 Current Standards

ISO 7498-2, *Information Processing Systems—Open Systems Interconnection Reference Model, Security Architecture.*

ISO/IEC 8613, *Information Technology—Text and Office Systems—Office Document Architecture (ODA) and Interchange Format.*

CCITT X.509, *Message Handling System, ISO/CCITT X.400 Directory Authentication Framework.*

ECMA CMA 138, *Security In Open Systems—Data Elements and Service Definitions.*