Chapter 1

Introduction

R/3 is an integrated system designed for national and international business use. The software package is implemented by companies working in one country and requiring one language as well as multi-national, multi-language organizations. These multi-national companies do business throughout the globe. Their business transactions thus may cross national language boundaries.

This document presents technical information on installation and administration as well as guidelines, and rules to follow for the implementation and application of a single central R/3 system in such a multi-national language system (MNLS). The described functionality for a multi-codepage capable R/3 System is implemented in R/3 release 2.2F and 3.0 (and later).

Business Scenario

Multi-national organizations may be using several distinct R/3 Systems per country or country group and these systems are periodically consolidated in a separate consolidation system. However, a multi-national organization may have the need for all of its organization units such as company codes, plants, or sales organizations to be operating in one single R/3 System. This scenario may be chosen, for example, by organizations that follow a global business strategy and manufacture and that offer the same products world-wide.

This implies for the technical environment that one central R/3 database simultaneously services all business application processes in various countries. Furthermore, end-users may have the need to access the R/3 System in their local languages and process transactions involving business partners who require the support of languages other than their own local language. There are also communication languages used during internal and external communication with business partners.

In order for R/3 to support this multi-lingual business, the system must be able to recognize and correctly process transactions and data containing a mix of languages. The R/3 System language is designated at the time the user logs onto the system as well as by the character set (codepage) used by the R/3 frontend (SAPGUI) and application server. The relevant codepage support is provided by the operating system where the SAPGUI and the application server processes exist. A codepage defines the characters available in a given operating system. These characters in turn determine which languages are possible within which codepage.

Codepages

Codepage

In a system, a codepage (character set) defines symbols (characters), which can be used in programs and which can be displayed in output devices (printers, terminals, ...). Three types of codepages exist. These differ in the identification of a single symbol: single- and double-byte codepages and Unicode.

Single-Byte Character Sets

In a single-byte codepage, each symbol is specified with a single byte (see Appendix A.1). The first part of a codepage table (0x20 - 0x7E) essentially contains printable characters of the ASCII character set (English alphabet)[1] and the second half contains country-specific symbols (e.g. German Umlauts). The single-byte codepage *Latin-1* (ISO 8859-1, see Appendix A.2) defines the symbols used in most Western European countries.

Double-Byte Character Sets

Some languages have so many symbols (Asian languages, e.g. Japanese[2]) that the complete character set can not be described with one byte (256 characters) (see Appendix A.3). The double-byte character sets identify a character either in one or in two bytes. For example in the Japanese codepage (EUC), if the first byte has a value between 0x81 and 0x9F (most significant bit), then the current character is specified in two bytes. Otherwise, one byte identifies a symbol. Examples for double-byte codepages are *EUC* (Extended UNIX Codeset) or *S-JIS* (Shifted Japanese Industry Standard)[3].

Unicode Character Set

Unicode is a so-called wide-character codepage. We examine the two-byte Unicode, which is defined in the ISO 10646 standard. All characters are identified with a two-byte value. Therefore, up to 65536 symbols can be specified in the codepage. Unicode eliminates the problem that two symbols from different codepages have the same hexadecimal value.

Others

As well, there are other schemes of encoding (like JIS). But those cannot be used by the R/3 System internally, they can only be used for output.

---

[1] The area 0x00 - 0x1F (before the ASCII signs) contains non printable control symbols.

[2] Japanese consists of the following types of phonetic characters: Kanji, Katakana, Hiragana.

[3] For the codepages EUC and S-JIS exist different implementations, which code different character sets (languages).

Codepage-Dependent Components in the R/3 System

The following list contains some exemplary types of data in the R/3 System that are codepage-dependent:

- texts (help texts, CUA menus, etc.)
- addresses (in the master data)
- key fields in database tables
- lists in optical archives
- ABAP/4-Programs that use EXPORT/IMPORT (local program data is stored in a global R/3 area (EXPORT) and can be read by other applications in the R/3 System (IMPORT).)

The following list describes some of the components in the R/3 System that are codepage-dependent.

- C programs that use standard string functions (e.g. isdigit() or strlen()). These functions expect the character size of one byte and assume that the same codepage is used as in the creation of the string.
- The implementations of NLS[4] (National Language Support), S-JIS or Mandarin differ on the different platforms (Windows NT, UNIX, ...)
- The sorting order of table entries currently follows the order of the symbols in the codepage table. Future releases (3.0 and later) allow the R/3 applications to influence the sorting order with additional criteria (e.g. sorting of German Umlauts).

[4] Definition of country-specific signs in strings: definition of the codepage, definition of the decimal point form, or definition of the display pattern of time and date