Provide accurate and consistent language support for your global MDMP systems by converting to Unicode

by Alexander Davidenkoff

Alexander Davidenkoff, Ph.D.
Solution Manager,
Globalization Services,
SAP AG

After receiving a doctorate in electrical engineering at the University of Karlsruhe, Alexander Davidenkoff joined SAP AG in 1992 to collaborate with companies in Eastern Europe, Russia, and the Asia-Pacific region. He now works as a solution manager for Unicode rollout, time zones, global integrated solutions using SAP NetWeaver, complex systems architectures, and related topics. He supports, advises, and trains colleagues, clients, and partners across the globe. You may reach him at alexander.davidenkoff@sap.com.

Suppose you have an SAP table containing a text field that has a byte with the hex value E6. What character does that code represent? The answer depends on the *language* context in which that particular character will be used. Supporting languages with different character sets on a single system has often involved complicated — if not downright impossible — configuration and programming efforts. Since the early days of commercial usage of computers, consumers have needed a standard, cross-platform approach to representing characters properly, regardless of the language context in which the character is used. As early as the 1990s, multinational companies needed to run multiple businesses, using multiple languages, in a single SAP R/3 system. This requirement has only intensified since 2000 with globalization and the proliferation of business systems (most SAP customers now have more than one system in their SAP Business Suite, not to mention their SAP NetWeaver landscapes).

Unicode,[1] which emerged in the early 1990s, provides the de facto solution for a cost-effective, consistent, and standardized approach for language support in your global systems. Unicode provides a unique number for every character, no matter what the platform, the program, or the language. In doing so, Unicode remedies the "mess" made by previous highly restrictive workaround solutions running multilanguage data in one system. The underlying source of the trouble was that these systems reused the same binary codes across different character sets of languages, also known as "code pages." A code page defines the set of binary or hexadecimal values used to represent a given character set. For systems using a single byte for one character, you can define a maximum of 256 different characters — just enough to represent the character set of all Western European languages. However, you may also need to represent additional character sets, such as the Slavic languages (Russian, Ukrainian, and Bulgarian, for

[1] For more information about Unicode, visit the Unicode Web site at www.unicode.org.

example), that use a Cyrillic character set. If using only a single byte for both the Cyrillic character and Western European language characters, you will be faced with a dilemma: You have to use the same binary codes for the special characters of the Western languages that you use for the Cyrillic languages. For example, the German special character "ö" uses the hexadecimal code E6 on the Microsoft (MS) Windows code page 1252 for Western European languages, but the Cyrillic letter "ц" uses the same code E6 in the MS Windows code page 1251 for Cyrillic-based languages. So, how do you distinguish these characters when you run a global system that supports both German and Russian? Making matters worse, the code page for one character set may not always be unique. That is, there can be several code pages with different binary codes for the same character set on different IT systems. For example, for Cyrillic languages, there are the code pages MS Windows 1251 Cyrillic, IBM PC 866 Cyrillic, ISO-8859-5 Cyrillic, and still other code pages that use different binary codes for the same Cyrillic characters; therefore, exchanging the data of one language between different IT systems that use different code pages for the same language complicates the conversion of multiple languages to Unicode even more. Because Unicode assigns a unique code to each character, regardless of its language — the German character "ö" has the Unicode code U+00F6[2] and the Russian character "ц" has the Unicode code U+0446 — the distinctive characters can be combined easily in IT systems with the Unicode code page.

For SAP teams worldwide, in particular those that run a mixed code page Multi-Display/Multi-Processing (MDMP) system, the need to convert to Unicode is inevitable.[3] The ultimate goal of a Unicode conversion is to convert all of the character and text data in your system to Unicode format. Because all data is stored in databases, every field of every database table that carries language-specific character and text data[4] must be converted from a non-Unicode format to a Unicode format. How can you migrate to Unicode without a major business interruption? The answer to this question depends on the SAP system version, code page configuration, database size, number of custom programs and interfaces, etc., so you first need to consider the following:

- **New SAP installations:** New releases of SAP NetWeaver and SAP NetWeaver-based applications that are released in 2007 or later will only support new installations of Unicode systems. The installation procedure of an SAP Unicode system is identical to previous non-Unicode new installations.
- **SAP systems running on releases prior to SAP NetWeaver 7.0[5]:** Since SAP Web Application Server (SAP Web AS) 6.20, all SAP applications are supported in Unicode. Therefore, it is strongly recommended that non-Unicode SAP systems be converted to Unicode as soon as possible, particularly systems that support blended code page or MDMP configurations[6] because they have many restrictions. See the sidebar on the next page for more information on pre-Unicode SAP systems.
- **Upgraded SAP systems running on SAP NetWeaver 7.0 or ERP 6.0[7]/ECC 6.0:** Because SAP NetWeaver 7.0 and SAP ERP 6.0 no longer support MDMP systems, SAP offers special procedures for a combined upgrade and Unicode conversion for software releases before SAP Web AS 6.20.
- **Non-Unicode SAP single code page systems:** SAP NetWeaver 7.0, SAP ERP 6.0, and other

[2] The notation for a character in the Unicode Code Chart is U+<*hexadecimal value*>; for example, "U+00AE" represents the "Registered" symbol (®).

[3] As of SAP NetWeaver 7.0 (formerly SAP NetWeaver 2004s), which underlies SAP ERP 6.0 (formerly SAP ERP 2005), Unicode is the only supported option for SAP systems using languages with special characters and mixed character sets. SAP recommends all SAP customers migrate to Unicode at the next opportune time (e.g., during an upgrade). This recommendation comes out of necessity: The new European Euro currency symbol "€," for example, is only supported by Unicode.

[4] Numeric and binary data do not have to be converted because they do not contain language-specific characters.

[5] Formerly SAP NetWeaver 2004s.

[6] This article discusses MDMP systems only. Unambiguous blended code page systems are treated like single code page systems. Ambiguous blended code page systems can be seen as predecessors of MDMP and are converted to Unicode in a very similar way. You can find information on the possible differences in specific SAP Notes and documentation.

[7] Formerly SAP ERP 2005.

Pre-Unicode SAP systems

Prior to using Unicode* as its standard for global language support, SAP supported character sets for over 30 languages. The code pages utilized were of four different types:

- **Single standard code pages** support specific sets of languages. The number and combination of languages that are supported cannot be altered.
- **Unambiguous blended code pages** are SAP proprietary code pages that contain characters from one or more standard code pages. This type of system increases the combinations of languages that can be used. Functionally, an unambiguous blended code page system uses a single code page.
- **Ambiguous blended code pages** are SAP proprietary code pages that contain characters from several standard code pages and overlap some of their binary encodings. This type of system, which can be seen as the predecessor of an MDMP system, increases the combinations of usable languages.
- **Multi-Display/Multi-Processing (MDMP) code pages** allow dynamic code page switching on the application server. This switching capability allows any combination of standard code pages to be used on one system with restrictions. The logon language of the user determines the switch to the correct code page so the user can work only in the code page in his or her dialog session. One potential issue may occur with other work process types such as background, update, spool, etc., because many users share these processes. Therefore, additional measures are needed. For example, a Russian user can only start a background job whose program runs in Russian, and so on.

* For a general introduction to character code pages, blended code pages, MDMP, and Unicode, refer to Michael Redford's SAP Professional Journal articles, "Globalizing Applications Part 1: Pre-Unicode Solutions" (September/October 2001) and "Looking Forward to the Unicode Advantage: Internationalization and Integration" (January/February 2002).

SAP Business Suite components still support SAP single code page systems; however, there are limitations, most notably when using a combination of ABAP- and Java-based applications and when communicating with external Unicode-based systems. Because more and more IT vendors are shipping their solutions as Unicode-only, single code page systems have become outdated. SAP recommends converting single code page systems to Unicode as well.

Even if your current business is based on only single character set languages, you need to consider converting to Unicode and potentially upgrading to more current releases of SAP. The need for new languages may be upon you sooner than expected as you expand your business to new markets and countries. For instance, consider China and its booming economy — in order to take advantage of this opportunity, you need to have Chinese language support in your SAP systems and hence the need for Unicode.

This article presents an overview of a Unicode conversion and in particular the special treatment necessary for the conversion of MDMP systems to Unicode. You will learn what is involved in a Unicode conversion, what you need in order to get ready for the conversion (evaluating system and hardware needs, for example), and some of the basic terms and concepts. The focus here is on the automatic and manual language assignment scans and tools needed to prepare mixed code page text data before and after the conversion to Unicode. You will be introduced to

specific solutions provided by SAP for the special circumstances that surround SAP systems that are not yet on a Unicode-supported SAP release and therefore need to be upgraded before or with Unicode conversion. You need this information in particular for the combined upgrade and Unicode conversion of an MDMP system to SAP ERP 6.0/ECC 6.0 in which MDMP is no longer supported. Although this article concentrates on the Unicode conversion of the SAP ERP components family, other SAP Business Suite components can be converted to Unicode in a similar way.

Project managers, consultants, and administrators who are planning and performing a Unicode conversion of an MDMP system, alone or as part of an upgrade to SAP ERP 6.0, will find the information presented here invaluable. Understanding SAP Basis and SAP NetWeaver is important and an understanding of languages and their alphabets and scripts is helpful. There's a lot to cover, so let's get started.

> ***Note!***
>
> The topic of Unicode conversion is vast. This article does not cover every aspect, but rather provides information that you can use to begin the process. Having a firm understanding of what is involved in this type of project will help you start on solid footing.

Getting ready for your Unicode conversion project

Unicode conversions require careful planning and preparation because the project has unique features. This is particularly true if you have to convert an MDMP system to Unicode.

In preparation for your Unicode conversion, you need to:

- Assess your current system landscape and component releases
- Evaluate hardware needs
- Get appropriate guides and documentation

Assessing your current system landscape and component releases

First, you need to check the current SAP components and releases running across your SAP system landscape to determine if it or parts of it support Unicode. This will help determine the path you need to follow to move your system landscape to Unicode:

- If you are running a release prior to SAP Web AS 6.20, you must first upgrade to a Unicode-supported release. **Figure 1** provides a list of minimum target releases. SAP offers combined procedures for upgrading and converting to Unicode together — these procedures are covered later in this article. If you have an MDMP system and plan to upgrade directly to SAP ERP 6.0, you should use one of these combined procedures.
- If you are running SAP Web AS 6.20 or higher, you can convert your system to Unicode directly because all of your SAP applications are supported in Unicode.
- If you are running any of the SAP Business Suite components, you need to know if your component supports Unicode. **Figure 1** shows the earliest releases[8] that support Unicode.[9]

Evaluating hardware needs

Because you want to run a Unicode system with the same performance characteristics and transaction volume as before the conversion, additional hardware may be required — this is especially necessary when system memory is of great importance.

[8] See SAP Note 79991 for further details.

[9] For more information on individual releases and platforms that support Unicode, go to the SAP Service Marketplace at http://service.sap.com/pam; for z/OS support, see SAP Note 728743 for details.

Component	First Unicode-supported release
SAP ERP (R/3 Enterprise)	SAP R/3 Enterprise Extension Set 2.00 SAP R/3 Enterprise Extension Set 1.10
SAP CRM	SAP CRM 4.0
SAP SCM	SAP SCM 4.x
SAP SRM	SAP SRM 4.0
SAP Solution Manager	SAP Solution Manager 3.1
SAP NetWeaver	SAP BW 3.5 SAP SEM 4.0 SAP Mobile Infrastructure 3.0 SAP XI 2.0 (Unicode only) SAP EP 6.0 (Unicode only) SAP MDM 5.5 (Unicode only)

Figure 1 Earliest Unicode-supported releases of selected SAP Business Suite components

CPU	**RAM**
• +30% • Depending on: - Existing scenario - MDMP, double byte usage	• +50% • SAP application servers based on UTF (for example, UTF-16)
Database size • UTF* – depends on database vendor - UTF-8 (up to +10%) - UTF-16 (+30% to 60%)	**Network load** (SAP GUI for Windows) • UTF-8 format • Almost no change because of efficient compression
* Unicode Transformation Format (UTF) is an encoding form (or schema) that assigns each Unicode value to a unique code unit sequence. The Unicode standard defines the formats UTF-8 and UTF-16 for use in SAP systems. For more information, go to the glossary of Unicode terms at http://www.unicode.org/glossary.	

Figure 2 Average additional hardware requirements for SAP Unicode application servers

Figure 2 provides general SAP recommendations based on parallel benchmarking between Unicode and non-Unicode systems under the same conditions. It shows the additional *average* Unicode hardware requirements for the SAP application servers compared to non-Unicode systems with the same performance and volume. For example, you will need to add an average of 50% additional memory to each application server in a Unicode system to achieve the performance and volume of a non-Unicode system.

In the case of standalone database servers, contact your database vendor. You will also want to work with your hardware vendor to determine more precise sizing requirements.

If you have to upgrade to a Unicode-supported release first, or you choose an integrated procedure for upgrade and Unicode conversion, then combine the requirements for both the upgrade and Unicode conversion in order to address the required hardware needs once.

Getting appropriate guides and documentation

Before you begin your Unicode conversion project, you will need the most current documentation, which includes the following:

- **Unicode Conversion Guide**: This guide must match your SAP code page configuration — single code page or MDMP — SAP NetWeaver release, and support package. You can find this guide at the SAP Service Marketplace at http://service.sap.com/unicode@sap.
- **SAP Heterogeneous System Copy Guide:** This guide provides important information on database export and import activities. You can find this guide at the SAP Service Marketplace at http://service.sap.com/instguides.
- **Installation Guide:** This guide describes the required SAPINST and R3load tools and how to install a new Unicode system. You can find this guide at the SAP Service Marketplace at http://service.sap.com/instguides.

If you combine an upgrade with the Unicode conversion, you will also need the following:

- **Upgrade Master Guide:** This guide must match your target release and is found at the SAP Service Marketplace at http://service.sap.com/instguides.
- **CU&UC (Combined Upgrade and Unicode Conversion) and TU&UC (Twin Upgrade and Unicode Conversion) Guides:** See SAP Note 928729 for CU&UC, SAP Note 959698 for TU&UC, and http://service.sap.com/unicode@sap for more information.
- **Installation Master CDs/DVDs:** The CDs/DVDs must match your Unicode release. You must have the CDs/DVDs for the new installation of your non-Unicode and Unicode SAP systems. The installation master CD/DVD contains the SAPINST tool and related material.

Always check for new SAP Notes from the SAP Service Marketplace at http://service.sap.com/notes.

Unicode conversion basics

There are a few key terms, tools, and concepts you need to know before beginning a Unicode conversion project:

- **Converting a non-Unicode database to a Unicode database:** The conversion procedure from a non-Unicode to a Unicode database is done in a very similar way to an SAP heterogeneous system copy or OS/DB platform migration. You export the non-Unicode database to dump files using the SAP tools SAPINST[10] and R3load. During the *database export*, R3load converts every character of the non-Unicode database to the Unicode format with two different conversion procedures: one for a single code page system and another for an MDMP system. You reload (*database import*) the dump files into the new Unicode database. The real conversion of the text data happens (with a few exceptions[11]) during the export process.
- **Single code page system:** If you need to convert an SAP single code page system to Unicode, you're lucky: Each character is already represented by a unique binary value in the database and can be easily converted to the correct corresponding Unicode code. The Unicode conversion of a single code page system is therefore straightforward. In addition to the database conversion, using suitable optimization methods for low downtime, the main work and preparation involves the adjustment and testing of custom-developed programs and interfaces for the new ABAP syntax (Unicode enabling) and a few other steps, which you will find in the Unicode Conversion Guides.
- **MDMP system:** Because of the mix of several code pages, the MDMP system conversion scenario requires special preparation and the presence on your project team of language experts who read and speak each of the languages currently used in your system(s).

[10] Refer to the "Unicode Conversion Guide" and "SAP Heterogeneous System Copy Guide" for more information.

[11] Discussion of these exceptions goes beyond the scope of this article, but you will find all details in the guides and documentation.

An MDMP system contains many texts that have the same binary codes in the database but belong to different language code pages. Therefore, you cannot simply export and import the MDMP database with Unicode conversion as you can in the single code page system scenario. Instead, you must first identify to which language and code page the text data belongs.

Note!

In general, the conversion of an MDMP system to Unicode is significantly more complex than a single code page system. For both scenarios, you will need SAP Basis and technology experts who are familiar with SAP heterogeneous system copies and who ideally have a certification for OS/DB platform migrations.[12]

- **ABAP Unicode enabling of customized programs and interfaces:** To support ABAP development in SAP Unicode systems, SAP has enhanced the ABAP programming language leading to a new ABAP syntax with modified, extended, and new ABAP language declarations and statements (as of SAP Web AS 6.10/6.20). In a non-Unicode SAP Web AS 6.20-based system, you still have the choice between the old and new ABAP syntax, but when you start your Unicode system for the first time, you must have the new syntax in place. Therefore, you need to adjust all of your customized programs and interfaces to the new ABAP syntax, a process referred to as *ABAP Unicode enabling*.[13] There are special tools that help you with these adjustments, which are thoroughly explained in SAP guides and documentation about ABAP Unicode enabling.

Preparing your MDMP system for Unicode conversion

As with most, if not all, projects, preparation is critical in regards to the conversion of data to Unicode. To prepare an MDMP system for a Unicode conversion, you must first get the text data in your database ready. Depending on the size of your database, number of active code pages and languages in your MDMP system, number of production applications, and the high or low usage frequency of multilingual texts in application data, preparing your database can be time-consuming and labor-intensive.

Tip!

To find out your code page configuration, run the program RSCPINST in transaction SE38 or SA38 and select Simulate, or display the entries of the table TCPDB with transaction SE16. If there is exactly one code page number (first digit not 6), then you have a single code page system. If you find several entries, you have an MDMP system, and the table field TCP0D-COUNTRY should be empty. If the first digit in TCPDB starts with 6, you have a blended code page system.

[12] Unicode conversions are defined as a homogeneous system copy and do not require a migration key or a certified migration consultant, but the use of such consultants is nevertheless highly recommended.

[13] For more information on this topic, refer to the following *SAP Professional Journal* articles: "Looking Forward to the Unicode Advantage: Internationalization and Integration" by Michael Redford (January/February 2002), "File I/O with ABAP — Problems, Workarounds, and Prudent Practices" by Gerd Kluger (November/December 2001), and "Improve Testing by Tracing ABAP Program Execution: Take a Closer Look with the Coverage Analyzer" by Christian Hansen (September/October 2002).

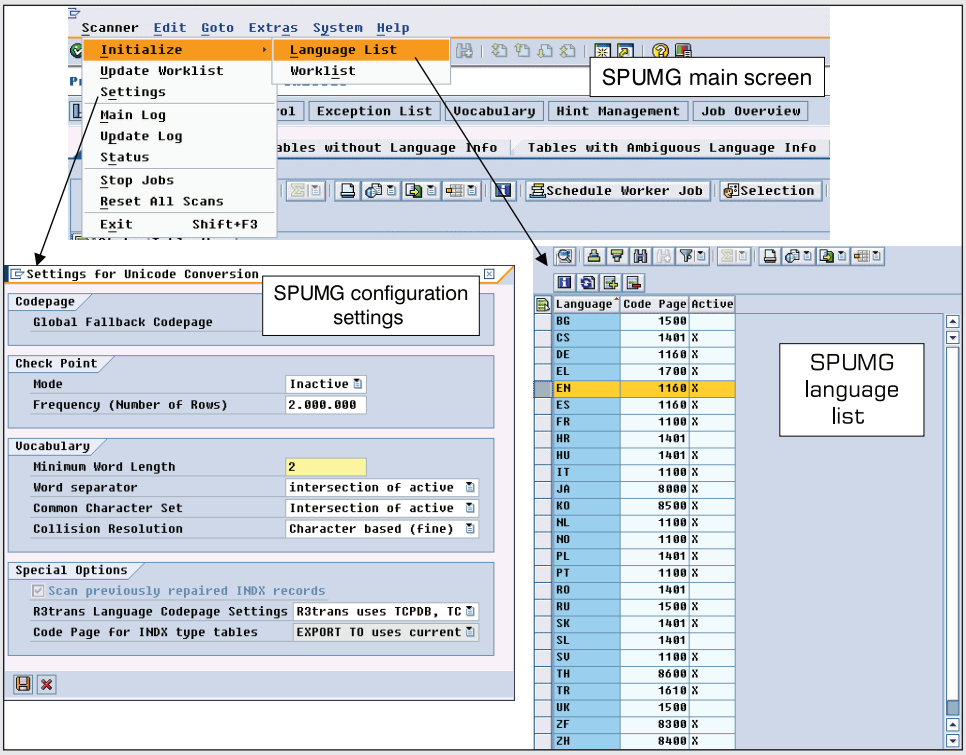

Figure 3 Configuration and initialization of transaction SPUMG (SPUM4 looks very similar)

To prepare your database for conversion, you need to run several *scans* that analyze the text data and assign languages to it. Although the goal of the scans is to assign languages to as much of the data as soon as possible, there will still be data left unassigned. This unassigned data resides in special tables in the database referred to as a *vocabulary*. There are more scans that you can run to assign languages to the data in the vocabulary. To facilitate the processing of the vocabulary, you can designate specific language conditions using *vocabulary hints*. After processing the vocabulary there will still be unassigned data, but now that the automatic analyses of the data are exhausted, you need a team of language experts to manually assign languages. Finally, you will run the *reprocess scan*. This scan completes the language assignments and simulates the Unicode conversion.

Let's take a closer look at preparing your database for Unicode conversion.

Automated database scans for building the vocabulary

To scan your MDMP database, use transaction SPUMG (or SPUM4 in SAP R/3 4.6C). These transactions automate the analysis of the database and the building of the vocabulary.

To begin, you need to specify some initialization and configuration settings for your scans. As shown in **Figure 3**, you need to identify the global fallback code page[14] — shown in the example to be the SAP code page 1160 for MS Windows code page 1252. You need to define a minimum length of a word (the

[14] The global fallback code page is part of the SPUM4 and SPUMG initialization and one of the existing (that is, configured) MDMP code pages. You specify the global fallback code page as the default code page for the Unicode conversion if no dedicated code page information is available.

default is 3 bytes, but for systems with Asian languages, the recommended length is 2 bytes). The Unicode conversion guides outline several other settings — the discussion of which goes beyond the scope of this article.

You need to select the languages and code pages currently active in your database. As you can see, there is a language list that displays all active languages with an "X." There can also be inactive languages (without an "X") that exist in your system that will not be used for processing your language data. It is very important that you maintain an accurate language list from the beginning. Before you start the scans, you need to find out if your end users maintain language data for inactive languages. If so, you must set these languages to active so they will be included in the language processing.

You may have noticed in **Figure 3** that the LATIN-1 languages DE, EN, and ES are associated with SAP code page 1160, whereas the other LATIN-1 languages are associated with code page 1100. What is the difference? You need to specify code page 1160 for MS Windows 1252 if your end users work with special MS 1252 characters, which are in the hexadecimal range 0x80-0x9F. Code page 1100, which is the SAP default, does not include this range. The decision to use 1160 or 1100 depends on the behavior of your end users.

After you have specified the initialization and configuration settings, you start the SPUMG (or SPUM4) *worklist*. The worklist is a special table, which is generated automatically, that contains all SAP tables that need to be scanned.

Your database will be scanned in several phases in background jobs, which can run in parallel for faster processing. There are several scans that can be run, each with a specific purpose:

- **Consistency Check:** This is the first scan you need to perform. It is mandatory for both single code page and MDMP systems. Its task is to make sure that all your SAP tables are consistent — that is, that all SAP Data Dictionary (DDIC) table definitions are consistent with the underlying database tables and that all database tables are without any errors — so that no problem arises later when you convert the table to Unicode. This scan sorts the tables into three categories, called TabCats:
  - TabCat1 contains tables with language information.
  - TabCat2 contains tables without language information.
  - TabCat3 contains tables that contain ASCII text data only.
- **Tables without Language Information:** This is a critical scan for MDMP systems only; instead of analyzing every single character in the database, it uses a word as the smallest unit. A word is defined as an arbitrary string of characters enclosed by spaces or other delimiters. In this scan, every table in the database is analyzed to identify all words without any language code page information, and then the words that have no language code page information are written into the vocabulary. Identical words in multiple tables are registered once in the vocabulary with information about all tables where they occur. Note that this scan can run for a very long time, up to several days in large systems, but with multiple parallel background jobs, you can reduce the duration.
- **Tables with Ambiguous Language Information:** This next scan is required because users might not have followed all of the important rules in an MDMP system, such as consistently using the correct login language for maintaining business text data. For example, if the login language is English and the data maintained contains Chinese customization text data, all of the data can be stored in a TabCat1 table record with language key English. The records with language key English are expected to belong to the Western European LATIN-1 code page. In this case, however, records with the language key English can be Chinese; therefore, they will not have a unique language code page assignment and remain unknown. These records need to be treated like the records in TabCat2 tables — that is, as if they do not contain a language code page at all. By

running this scan, this unassigned data is moved into a TabCat2 table.

- **INDX Analysis:** This is another scan you must perform on MDMP systems. SAP INDX tables have a very special design: They store data in binary clusters and are processed in a different way from all other SAP tables during an MDMP Unicode conversion. You must perform this special scan to identify all INDX tables that contain words to be assigned in the vocabulary.
- **INDX Repair:** If the INDX Analysis scan inserts words in the vocabulary that need to be assigned to a language, you have to perform this scan to complete the special treatment of INDX tables. (You will find more details in the Unicode conversion guide.)

After these scans, the vocabulary has initially been filled with all words to be assigned to a language. To see how many unassigned words are in the vocabulary, you can access the number of total entries in the vocabulary monitor by selecting the Vocabulary tab in the SPUMG (or SPUM4) main screen (**Figure 3**).

To start with the language assignments, you need to consider running the following scan:

- **Tables with Language Information:** You need to run this scan, which scans the database and compares it with the vocabulary, if this is the first time in your system landscape that you've used SPUMG or SPUM4. Remember that any word in a table with language information is already considered "perfect" because the code page is known. This scan looks to see if the same word with an unknown language, which is already in the vocabulary from the previous scans, also appears in another table with language information. If it does, then the language information is also assigned to the word with an unknown language in the vocabulary. Of course, the language information assigned might not be correct so your language expert still needs to check the assignment. If you have an existing vocabulary in another similar system in your landscape then you should transport that vocabulary into this system with the import method that is described in the next section.

After these scans you will spend a good amount of time processing the vocabulary, either with tools-based processing or manual processing. You need to run these next two scans before your MDMP system database can be converted to Unicode. Let's take a look at them.

Tools-based language assignment

Before beginning the manual language assignment task, you can process the vocabulary using tools that will further analyze the unassigned data and help to identify the languages:

- **Transport of vocabulary from other systems**: If you have a processed vocabulary from another similar system in your system landscape from a previous SPUMG or SPUM4 transaction, then you can download or insert it into an SAP transport request. You can upload that vocabulary into the current system, and a special method transfers all language assignments for the words that exist as well as the current vocabulary. Because business data is usually similar across your system landscape, you will probably get a large number of words that have already been assigned. Now, you only have to work on the remaining differences between the systems.
- **Import of special SAP vocabularies:** In some cases, you may wonder where a word in the vocabulary originated from since your language team never worked with it. Every SAP system contains a certain amount of predelivered language data that is scanned together with your data and therefore inserted into the vocabulary. SAP delivers predefined vocabularies with the correct language assignments, which you can import into your system according to SAP Note 756535. Importing these special SAP vocabularies allows you to focus on your own business-specific language data.
- **Language guesses with statistics:** There is a "semi-automatic" method that guesses the language of a word based on a statistical pattern and assigns the language to the word in the vocabulary. There is another "semi-automatic"

method that guesses the language of unassigned words using special language statistics based on byte pattern distributions. Again, the method assigns the language to the word, but you need to make sure your language experts verify these types of assignments. The statistics are available for many SAP languages, and the quality of the guesses depends on various properties of the language and its combination with other languages in your SAP system. For more information, see SAP Note 756534 and the Unicode Conversion Guide.

- **Vocabulary hints:** These are language rules that you define to designate specific language assignment conditions. Hints are particularly useful for large tables and if you have unique business rules that govern the data that is entered in your database tables. For example, you might assign all customer addresses in a table with country key JP (Japan) as Japanese or all invoice texts in a financial table with company code 200 (which has been configured for Russia) as Russian, and so on. You can use hints to process the data in those tables instead of manually assigning languages, which saves time and effort. Note that you only have to create hints once in your system landscape because they can be transported and your other systems likely have similar business data.

After all the scans and tools-based processing of the vocabulary, the only way to evaluate the remaining unassigned data is to have language experts, native speakers, or translators perform the remaining assignments manually — they will make the final decision as to which language each word belongs. The ideal person knows the language and your "business dialect" because many words may be company-specific names, abbreviations, and business-related terms. Therefore the next step is to go through the vocabulary manually, looking at each word to further analyze its language.

Manual language assignment

A special obstacle present in an MDMP system is that you can only see the characters of one language code page (based on your logon language) correctly in your SAP GUI display. For example, if you log into the MDMP system in English (EN) and open the vocabulary, you can only see the words that belong to languages of code page LATIN-1 correctly because EN uses the LATIN-1 (SAP 1100) code page. As shown in **Figure 4** (on the next page), instead of a readable text, you see "garbage"; therefore, you know that these words cannot be LATIN-1 words. But what are they? To determine what this garbage is, you need to log in to your MDMP system in each of the configured language code pages on the system and see if any of the words appear correctly in that language. For example, if you know that the garbled text is probably Russian (RU), then once you use RU as your login language you should be able to see that the Russian words no longer appear as garbage, but rather as Russian words, as shown in **Figure 5** (on page 15).

Now your Russian language expert can identify which words are Russian and manually enter "RU" in the Language column (the "M" in the Filled by column identifies the user who assigned the language). This expert will also recognize that the word "LЖSCHEN," which was automatically assigned RU by the system, contains the Russian letter "Ж"

Note!

Manually evaluating the vocabulary is a tedious job. You have to scroll through a great deal of unreadable data, and you will almost never be able to identify all of the words. It is also possible that some words can be damaged because of improper usage in the MDMP system. You may also see "binary garbage" that comes from either hard-coded languages that have never been used but are in the system, or binary data that has been interpreted as character or text data. The vocabulary is client-dependent, so words from other SAP clients can appear in the vocabulary. Recommendation: Use as many hints as possible to shorten this job.

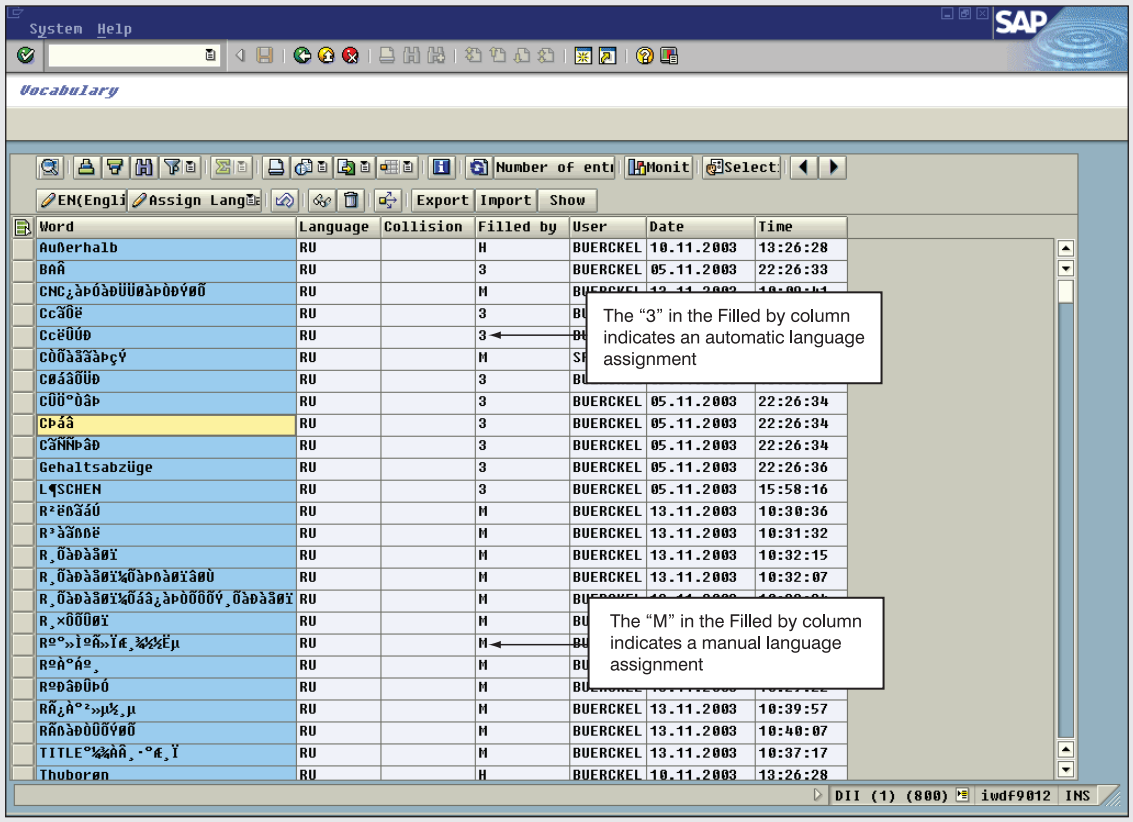

Figure 4 Effects of an incorrect English login language for vocabulary processing Russian text

but this is not a Russian word. In this case, the expert must ask the other vocabulary team members to help identify the word. The German (DE) language expert who uses DE (or EN) as the login language would identify "LÖSCHEN" as German and assign it to DE.

Your vocabulary team needs to log in using every reasonable code page that might apply and then go through the list searching for words in known languages.

Reprocess scan for processing a "secondary vocabulary"

After your language team has completed its review of the vocabulary and manual language assignments, there is still another major task needed to prepare for the Unicode conversion — finding the remaining words that are still unassigned in the database. These remaining unassigned words that are not in the vocabulary (e.g., words that are too short or too long), words that belong to a language that has not been configured as active, words referred to as *collisions*,[15] and words that for some unknown reason have yet to be identified. For this purpose, you perform the *reprocess scan.* This scan simulates the export of the database with its conversion to Unicode. During the database export process of an MDMP system, the

[15] Collisions are situations in which either the same word belongs to multiple different code pages (vocabulary collisions) or one table field contains words from multiple different code pages (code page collisions). Collisions either are not maintained or must be resolved manually.

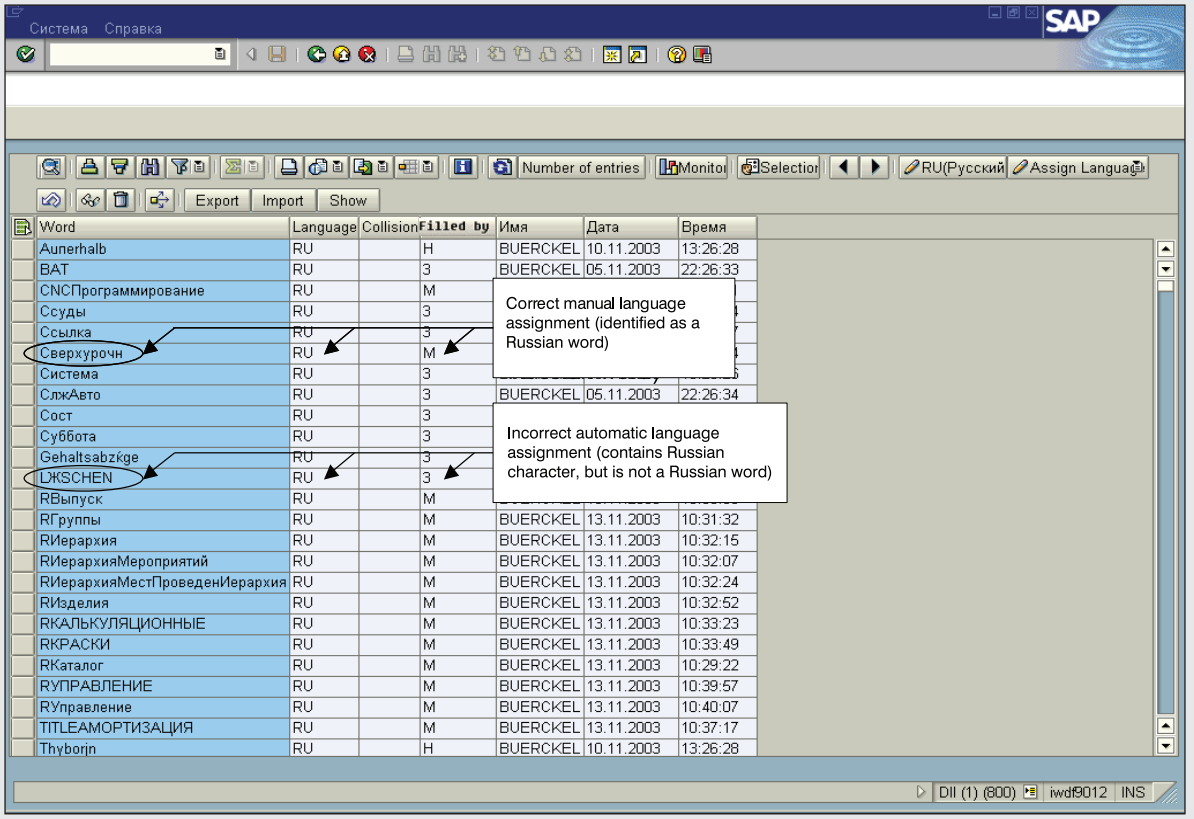

Figure 5 Effects of a correct Russian login language (RU) for vocabulary processing of Russian text

R3load tool looks in special control tables[16] for the language code pages assigned to each word and character in the database so the system can convert the data correctly to Unicode. If no information is available, the system uses the global fallback code page (specified during the configuration of the SPUMG or SPUM4 transaction) and writes a special warning in an additional XML log generated by the R3load tool.

During the reprocess scan, the system checks and writes all still unassigned or unidentified words to a *reprocess log* for each table, which is a kind of secondary vocabulary, in a different format. The reprocess log contains the table keys, the text records with the unassigned words, and a field for the language assignments. For each table, you can check the reprocess log to see which words or characters are unidentified. **Figure 6** (on the next page) shows the table FMCOBJC. The yellow triangle in the Status column and the "C" (that is, "Reprocess Log has been written") in the Message column indicate an unassigned word in the table. Clicking on the Reprocess Log button displays information you can use to find and analyze the affected word(s). For example, reason 8 (that is, "Word too short") indicates that the affected word is shorter than the configured minimum word length in the SPUMG (or SPUM4) configuration settings (**Figure 3**) and therefore was not stored in the vocabulary. If you look closely at the characters in the Text column, you can see that there is a section

[16] Note that for a single code page Unicode conversion every word in the entire database is converted using only one language code page without need for these special control tables.

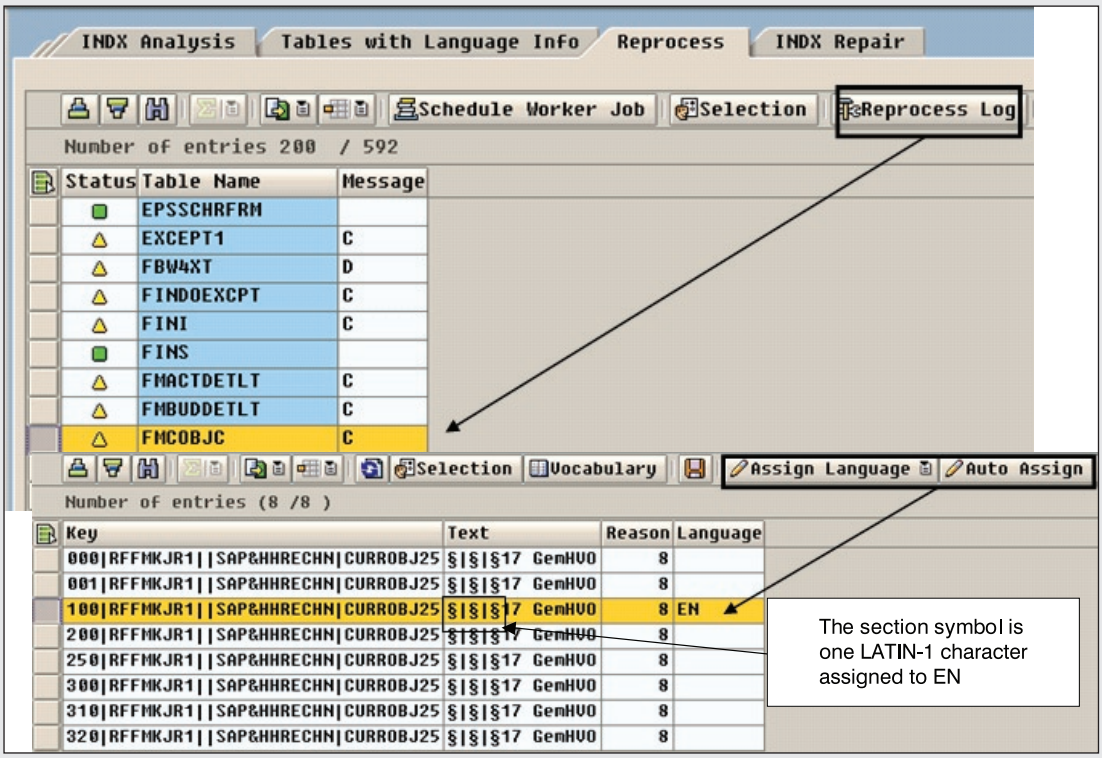

Figure 6 Reprocess log example

symbol (§) and a vertical bar (|) used as a field separator (assuming that you are logged in using the correct language). Because the § belongs to the LATIN-1 code page, the language EN must be assigned in the Language column.

> ***Note!***
>
> The reprocess logs contain entries from all SAP system clients. In the Key column, you can see the different client numbers. If you have words belonging to SAP client-dependent data, you must log in to every client, not only in your working client (e.g., log in to client 000, log in to client 100, etc.), to check and identify the data.

Each language expert needs to log in with the correct language code page to perform the language-assignment task. If you have a reprocess log from another system, you should first upload it into this system so that your vocabulary team has access to reprocess log entries that have already been assigned. Note that only one user can work on one reprocess table at a time.

Assigned reprocess log words are converted differently to Unicode than vocabulary words. While the R3load program converts the vocabulary words directly through the special control tables with the language assignment information during export, the reprocess log entries with their language assignments are written to R3load XML log files (also called reprocess, or repair, logs). So the standard R3load log files and the R3load XML log files are all part of the database export. Sometimes,

however, some of the data in the reprocess logs is incorrectly converted using the global fallback code page. After the Unicode conversion, you must still handle these incorrectly converted reprocess words. For this purpose, the XML reprocess and R3load log files are uploaded into the Unicode system and used in an automatic repair procedure.

Automatic and manual repairs after the Unicode conversion

After the database export and import into the new system has been performed,[17] there are some mandatory steps in the new Unicode system that you need to take before you can hand the system over to users, in particular the "repair step" in which words that have been incorrectly converted to Unicode are corrected. The conversion of the reprocess data from the scans before the Unicode conversion still has to be completed. You complete the reprocess data conversion in the new Unicode system using two repair methods provided through transaction SUMG — an automatic method and a manual repair method.

Automatic repair

All information for the automatic repair is contained in XML reprocess and R3load log files, which are generated during the database export with the R3load tool. After the new Unicode system is ready for login, you use transaction SUMG to upload all XML reprocess and R3load logs in a special format.[18] Tables that need to be repaired appear in the SUMG worklist with a red status icon and the status text "New – not yet being processed." To repair these tables, you perform an automatic repair by clicking on the Schedule Worker Job button in SUMG. These background repair jobs may run several hours. This is still part of the production downtime because many reprocess words are incorrect until the automatic repair has been completed.

After the automatic repair, you will still have data that could not be automatically repaired. As shown in **Figure 7**, that data is marked with a yellow triangle status icon and the status text "In Manual Processing," which means this data needs further analysis by your language team. Note that the tables with a green square status icon in the worklist are complete and do not need further processing.

> ***Note!***
>
> Automatic repair is a mandatory part of the whole Unicode conversion procedure to complete the reprocess data, while manual repair is only needed if the automatic repair was incomplete or if users detect unreadable or incorrect words in the Unicode system. Both repairs use the same transaction SUMG but in different ways.

Manual repair

You will probably never be able to process the vocabulary and reprocess entries 100% correctly, even if you have the best language experts on your team. So, in practice, you have to expect that some data will not be readable after the Unicode conversion because it has been converted with the wrong code page value or no language has been assigned in the vocabulary or reprocess logs. You need to perform a *manual repair* if the automatic repair could not complete all reprocess data or if your users detect unreadable words in the new Unicode system.

[17] I do not describe the actual conversion in this article, but you will find all the relevant information in the guides and documentation.

[18] For more information on automatic repair, refer to the Unicode Conversion Guide.

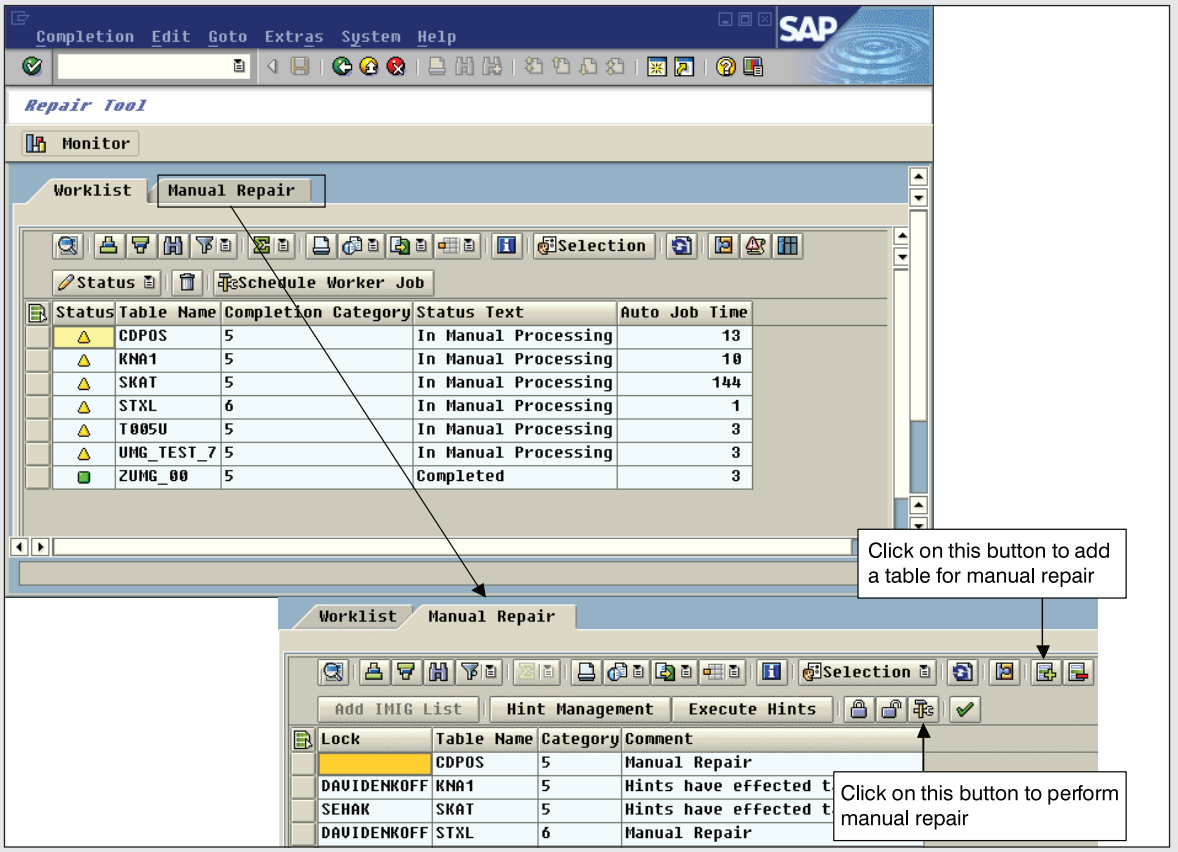

Figure 7 Repair tool in SUMG

Another important reason for the manual repair is the delta processing problem caused by new language data that is continuously created between the start of your vocabulary work and the Unicode conversion of your production system — this can be from a few days up to several weeks or even more — which does not appear in the vocabulary nor in the reprocess logs and remains assigned. This data is converted with the global fallback code page and is only correct if it belongs to this code page. In practice, however, many of these words will be first incorrect after the conversion and have to be manually repaired. There is no solution other than minimizing the delta processing time by reducing the manual repair volume.

> ***Note!***
>
> Incorrect conversions happen because language code pages in the vocabulary or reprocess logs have been assigned incorrectly or not at all. Unfortunately, there are no automatic ways to uncover language-assignment errors. Your end users will be the final test. Ensuring that the words in your databases are correct may be a lengthy ongoing process.

For example, assume again that the language

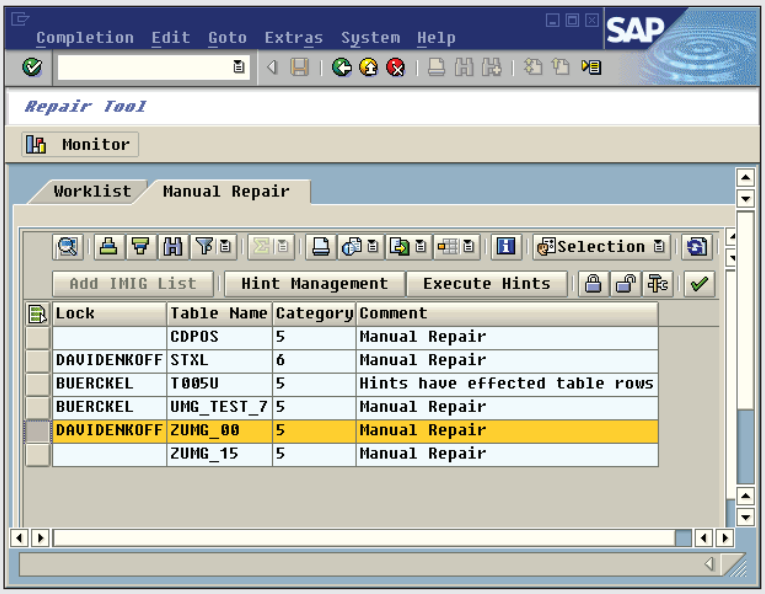

Figure 8 Tables listed for manual repair in the Repair tool

English (code page LATIN-1) was erroneously assigned to a Russian word in the vocabulary. A Russian end user would therefore encounter an unreadable word during the course of his or her job. The user would then ask for help from your language support team, who must find out why this word is unreadable and what happened with it during the Unicode conversion. Several time-consuming analyses might be required. In this example, the support team would find out that this word had been assigned to the wrong language (that is, English using LATIN-1). To repair the word, you convert the table field entry with this word back to non-Unicode with the misassigned language code page, in this case, English using LATIN-1. Then you convert the word again to Unicode, but now this time with the correct language code page — Russian. You can perform a manual repair in your Unicode system at any time, even if you discover incorrect words long after the go-live weekend. You do this with transaction SUMG.

The steps for a manual repair are described in the following paragraphs. Before you begin, you need to know whether the manual repair is necessary because of an incomplete automatic repair or because users have detected unreadable texts. Both cases are treated in the nearly same way, but here we'll assume that a user has found unreadable text and has requested a manual repair:

1. Use transaction SUMG, and check the Worklist tab to see if your table with words to repair is listed. If your table is not yet listed, add it by switching to the Manual Repair tab, as shown in **Figure 8**. Click on the Selection button and then choose the table from the database.

2. To repair a table, you first must lock it so that no other user can work on the table at the same time; if the table is listed, but is already locked because another user is repairing it, then you cannot work on it until it is unlocked. To lock a table, select the one that you want to repair (in the example, ZUMG_00), and lock it by clicking on the Lock button. As shown in **Figure 8,** your user name will appear in the Lock column (in the example, DAVIDENKOFF). This lets other users know the table is being repaired, and by whom.

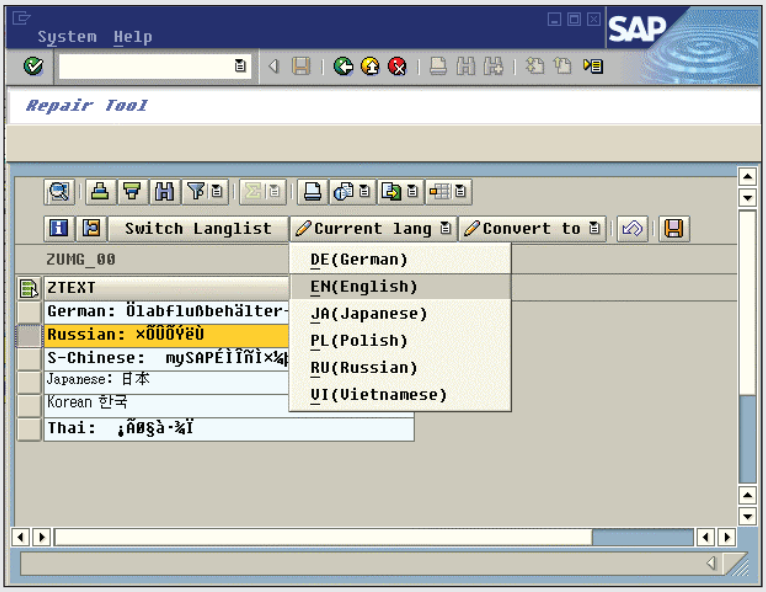

Figure 9 Assigning the incorrect language that was active during Unicode conversion

3. Once you have locked the table, click on the Repair button (**Figure 7**).

4. If you have a large number of table entries to repair, repair hints are recommended, which work in a similar way as the vocabulary hints in PUMG or SPUM4. They are script programs that can repair a large number of entries quickly. Click on the Hint Management button. In the dialog box that appears, create a condition in which you define a subset of the table with incorrect entries to avoid memory problems with very large tables. For example, let's assume that for some reason, all Japanese customer addresses in table KNA1 with selection field LAND1 = 'JP' (country Japan) are unreadable after the Unicode conversion. What you find is that the addresses have been incorrectly converted using code page 1100 for English (EN), rather than code page 8000 for Japanese (JA). To fix the problem, you can create a repair hint that selects all entries in table KNA1 with selection LAND1 = 'JP' that assigns the incorrect language EN, and then assigns the correct language JA. After you execute this repair hint, all Japanese customer addresses in the table are repaired and display in Japanese. Be aware that once you execute a repair hint, the repaired data is written into the application tables without an undo function. Note that you can transport repair hints to other systems so that you only have to create them once.

5. As shown in **Figure 9**, select a table row with an entry to be corrected and assign the incorrect language (i.e., the language code page) via the Current language button. Remember that the incorrect language is the one to which the affected table entries were (incorrectly) converted. Assigning the incorrect language might require some research in advance. In our example, we assume that a Russian word was incorrectly converted with the LATIN-1 language code page for English.

6. Set the correct language (i.e., the language with the correct code page of the original word) by clicking on the Convert to button, which performs the repair. As shown in **Figure 10**, the correct language is Russian. The marked entries are now corrected — i.e., converted back to non-Unicode with the wrong language code page (that is, English) and converted again to Unicode with the correct language code page (that is, Russian). The

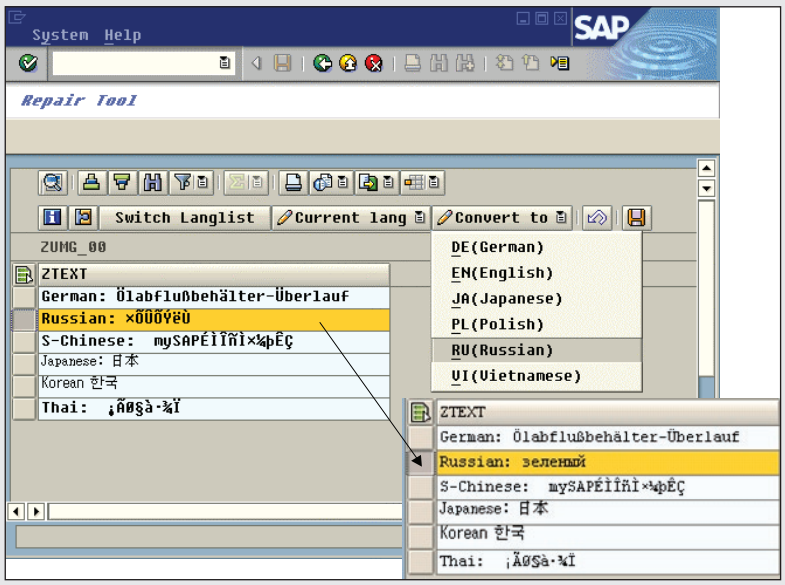

Figure 10 A manual repair in which the correct language is assigned by a language expert

entries are only written back to the original tables after you confirm and save the changes, therefore you can still undo an incorrect repair. If necessary, you may need to try different language code page combinations until you get the right texts, which should be validated by a language expert.

Best practices for using SPUMG, SPUM4, and SUMG

Before moving on to upgrades, I want to share some best practices for using transactions SPUMG, SPUM4, and SUMG when preparing your data for a Unicode conversion and repairing it after the conversion.

Using SPUMG or SPUM4

To make the most of the SPUMG and SPUM4 tools, consider the following guidelines:

- **Read the Unicode guidelines and related documentation:** Read all available SAP Notes and recommendations carefully. In particular, learn which automatic language assignment methods are available and for which languages they work well. These tools are continuously improved and should be reviewed by your language team on a regular basis.

- **Prepare the configuration of SPUMG or SPUM4 precisely:** Check to see which languages are actively used (or have been used) in your system landscape. All languages might not have been configured as login languages in the past, but end users might have switched their keyboard on the PC to enter business data in their local language (e.g., users may need to input Russian texts although the system is not configured for Russian). If you discover that there are languages that are used but not configured, add them to the active language list in SPUMG or SPUM4 settings. If you have Asian languages, SAP recommends that you set the minimum word length to 2 bytes because many Asian words consist of an ideograph (that is, a 2-byte Asian printed character) surrounded by spaces. If the default word length setting is 3 bytes, these words will not be added to the vocabulary. It is always better to see

an unassigned word in the vocabulary than in the reprocess log.

- **Run SPUMG on a recent copy of your production system and transport the vocabulary, hints, and reprocess logs:** After you have processed the vocabulary, created hints, and performed the reprocessing in one (the first) system, you should transport the vocabulary, hints, and reprocess data to all subsequent MDMP systems in your landscape so that you only have to perform a delta processing there. The transport of vocabulary, hints, and reprocess data into other systems is fully supported either by downloading them to local files or by creating a special SAP transport request; the correct choice depends on how your IT organization processes software changes.

  During the upload of the vocabulary into the subsequent systems, special mechanisms guarantee that only valid words specific to the subsequent systems are imported. Therefore, you should have a copy of the production system with real data; then, after processing, you reuse the vocabulary, hints, and reprocess data in all subsequent systems, such as development or quality assurance. Finally, close to the go-live weekend, transport the vocabulary, hints, and reprocess data into the production system and process the delta. Note that you must always run the database scans in every system that is going to be converted to Unicode, but you can transport the vocabulary, hints, and reprocess data between the various systems.

- **Allocate enough system resources and background processes for the scans:** The scan jobs can run very long — several days is not unusual — and should be run in parallel in several background jobs, each processing the next table in the queue. To facilitate the scan jobs, launch as many background jobs as you can afford, or run all jobs on a dedicated background server.

- **Run database and reprocess scans optimally:** Every MDMP system that will be converted to Unicode must be scanned. For the first time, run all scan jobs including the Tables with Language Information scan to populate the vocabulary with initial automatic assignments. Next, you need to process the vocabulary with all described automatic, semi-automatic, and manual methods, and then download it together with the hints and reprocess logs. For all subsequent systems, run the scans, but skip the Tables with Language Information scan. Instead, import (or upload) the vocabulary, hints, and reprocess logs[19] from the previous system.

- **Use vocabulary hints:** Before you start the manual language assignment, use vocabulary hints, especially if many words from large tables appear in the vocabulary, to significantly reduce the manual time and resource consuming language assignments.

- **Use language experts to manually assign languages:** There are different opinions in the SAP community. Some are convinced that even a single-language speaking person — typically the Basis consultant — can manage the manual assignment, just by "character guessing" or other smart ideas. This might work for languages with an "easy" character set, such as the Eastern European languages, but when you have Asian languages, do not try this — unless, perhaps, you can determine if an Asian-looking ideograph is Simplified, Traditional Chinese, or Kanji (Japanese). Even if you can, is this a correct word or just a random mixture of ideographs?[20] It is much better to let the language experts do this work. They can even be language students, who read and understand the language.

- **Complete the vocabulary as much as possible:** Even if you are not able to assign all words, you should try to complete the vocabulary as much as possible by assigning a default language to words that are still unassigned. You might apply an unused language (for example, use IS for Icelandic, if it is not one of the languages used in your database), so that you can identify these default assignments easily. Don't get

[19] For SPUM4, SAP Note 867193 describes how to download and upload reprocess logs.

[20] An ideograph is an Asian printed character.

nervous: Even if this default language is incorrect, you can correct the conversion in the Unicode system with transaction SUMG. With this default assignment, the affected word will not appear in the reprocess log.

Using SUMG

To make the most of the tools that the SUMG transaction provides, consider the following guidelines:

- **Plan enough production downtime for automatic repair:** Users frequently make the mistaken assumption that the Unicode conversion is completed after database import. The automatic repair with SUMG can take several hours depending on the number of reprocess entries and available hardware. Perform a realistic test of SUMG before production conversion.
- **Prepare your team for potential manual repair procedures long after conversion:** Incorrect text data is often found some time after the production system conversion because it is impossible to validate all text data before the go-live weekend. To perform such a repair quickly, the support team should be well trained in the procedures.
- **Use conditions and repair hints for manual repair whenever possible:** To avoid memory problems for large tables, always create conditions before you access a large table for manual repair. If you identify a huge set of incorrectly converted text data with a certain rule, create a repair hint. However, test the repair hint carefully before implementing it because repair hints perform direct updates on the application tables.

Procedures for combined upgrade and Unicode conversions

In June 2006, SAP released SAP ERP 6.0, containing many new features and integrated industry solutions. There will not be another major SAP ERP release until at least 2010. There are major global players still on SAP R/3 4.6C with an MDMP configuration and a large database who often want to upgrade to SAP ERP 6.0/ECC 6.0 directly. Because MDMP is no longer supported as of SAP NetWeaver 7.0-based releases such as SAP ERP 6.0,[21] the upgrade and Unicode conversion must be done together — how can you do both within a reasonable amount of downtime?

To allow both activities to occur together, SAP has developed combination procedures for upgrade and Unicode conversion in a single downtime — the *CU&UC* procedure (C*ombined Upgrade and Unicode Conversion*) and the *TU&UC* procedure (*Twin Upgrade and Unicode Conversion*). Both procedures enable you to perform an upgrade and a Unicode conversion within the same span of downtime. Let's look at each of these procedures.[22] Note that the simultaneous upgrade to SAP ERP 6.0 and Unicode conversion of MDMP systems is mandatory; in the case of a Unicode conversion of a single code page system, you have the choice to upgrade to SAP ERP 6.0 first, run the single code page system for a while, and then convert to Unicode.

Combined Upgrade and Unicode Conversion (CU&UC)

CU&UC makes it possible to upgrade an MDMP system from SAP R/3 4.6C to SAP ERP 6.0/ECC 6.0 and convert to Unicode in one continuous downtime, often during a (long) weekend. Because MDMP is no longer supported in ECC 6.0, this procedure is the only option if you have an MDMP 4.6C system and want to upgrade to ECC 6.0. The CU&UC procedure can also be applied to the earliest releases of SAP R/3 Enterprise and SAP ERP 2004.

If you use CU&UC, you start the upgrade of your SAP R/3 4.6C system as usual with the SAP upgrade PREPARE tool. There is, however, an additional step that asks you if you want to combine the upgrade with

[21] See SAP Note 7999, and contact SAP Globalization Services at globalization@sap.com for more details.

[22] For details, refer to the special CU&UC and TU&UC guides available at http://service.sap.com/unicode@sap and also see the SAP Notes.

a subsequent Unicode conversion. If the answer is "yes," then the tools for MDMP preparation with transaction SPUM4 are imported into your SAP R/3 4.6C system — these are nearly the same tools you use to prepare your MDMP system for a Unicode conversion in later releases. The SPUM4 activities can run during production and in parallel with other upgrade preparation activities; there are a few required synchronization points between MDMP and upgrade preparations. In particular, you create and process, or update, the MDMP vocabulary as you would in any release more recent than 4.6C. After you have completed the PREPARE and the SPUM4 activities, you start the upgrade with the SAPup tool, selecting the "Downtime Optimized" strategy (to minimize upgrade downtime in a production system).

The SAP upgrade SAPup tool adds a few Unicode-specific steps to the upgrade phases and runs with the SAP standard upgrade system switch method. Upon completion of the upgrade, you will have to perform a small set of required Unicode preparation activities. For example, you need to run a consistency check for new tables that have been added during the upgrade. The Unicode conversion can then start immediately; you perform and tune the database export and import in the same way as you did in a standalone Unicode conversion of an MDMP system. After the nearly identical post-processing steps as in a standalone MDMP case, particularly transaction SUMG (described earlier), the system is open for production.

There are two major downtimes, namely the upgrade downtime (after the upgrade phases for the *shadow instance*), and the database export and import. Although the upgrade downtime is relatively independent of the database size (assuming that all upgrade optimization methods are used), the time for Unicode export and import is dependent on the database size. Therefore, it is recommended that you invest in downtime optimization for the database export and import. The tuning methods and recommendations are the same as for any other Unicode conversions without an upgrade, so you should apply the same downtime optimization measures as you would in the single Unicode conversion case.

Although the main downtime step of the Unicode conversion follows the upgrade, several Unicode-specific preparation steps have been integrated into the upgrade preparation and execution to speed up those phases (compared with separate upgrade and Unicode procedures). You can now perform MDMP-specific scans and vocabulary processing in SAP R/3 4.6C using transaction SPUM4, which was ported from SPUMG in Unicode-supported SAP releases.

The ABAP Unicode enabling of customized programs and interfaces cannot be done in SAP R/3 4.6C systems, but rather must be done in a Unicode-supported release because it was not possible to realize the new ABAP syntax in 4.6C. Therefore, some additional activities are necessary:

- Upgrade a copy of the production system (sandbox) or development system to ECC 6.0 non-Unicode. You can also perform the Unicode conversion as a first test with this system.
- Adjust all customized programs and interfaces to the new ABAP syntax in this system.
- Include the adjusted programs in a transport that will be made available to subsequent upgrades.

In all subsequent upgrades, the adjusted programs will be automatically included, so upon upgrade completion they are Unicode-enabled, and the subsequent Unicode conversion can be directly performed The CU&UC steps are illustrated in **Figure 11**.

Note!

The ABAP Unicode enabling can only be done in the target release ECC 6.0 (or a Unicode-enabled SAP release); therefore, a sandbox or copy of the development system needs to be upgraded first.

Figure 12 (on page 26) summarizes the major steps of the CU&UC procedure, classified by downtime and uptime.

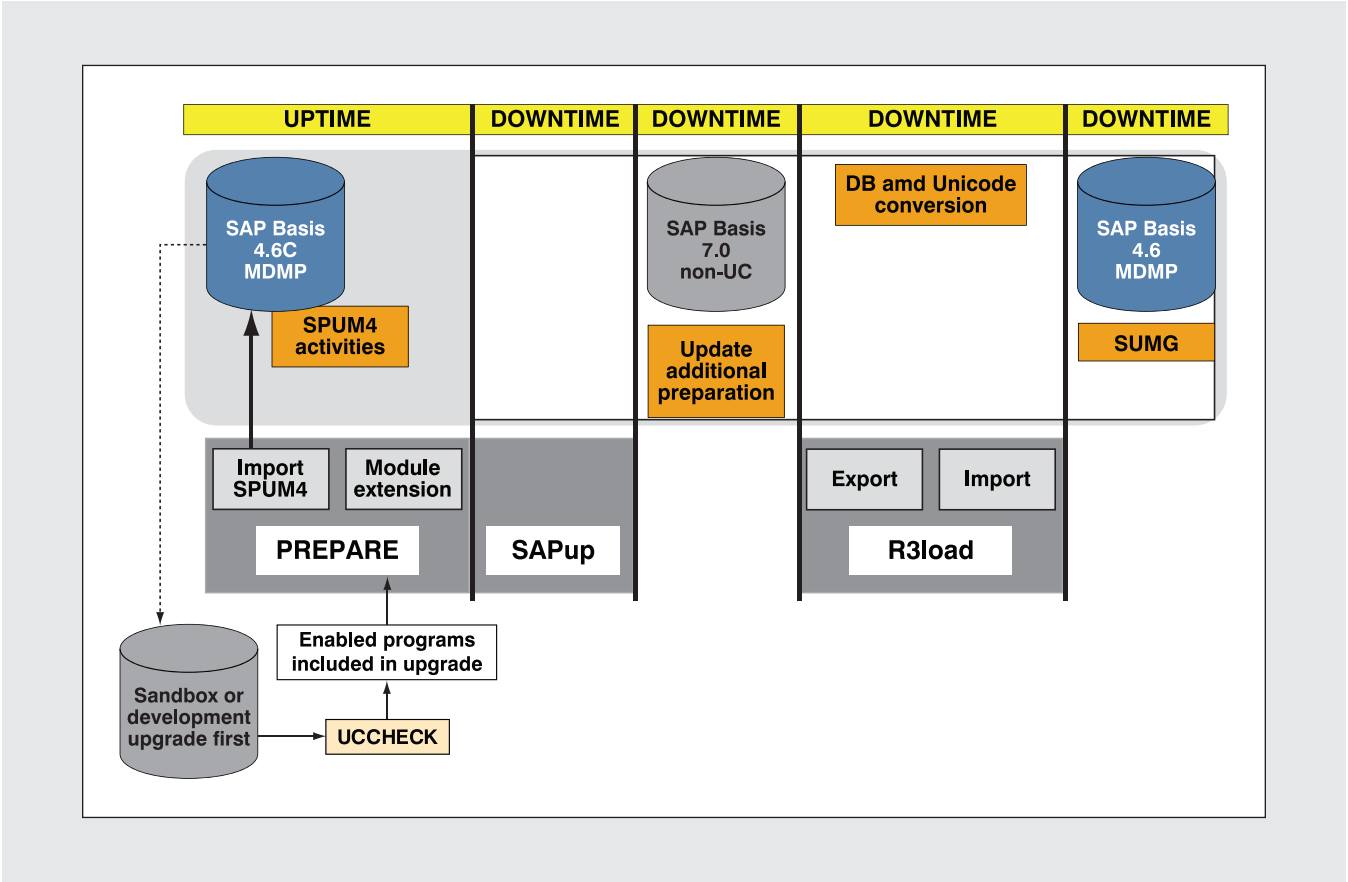

Figure 11 The CU&UC procedure

Combined Twin Upgrade and Unicode Conversion (TU&UC)

If you are on a system release that is earlier than SAP R/3 4.6C, and cannot afford to run the runtime-intensive SPUM4 database scans in the production system or want to upgrade to a Unicode-supported release prior to ECC 6.0 with a combined procedure, then there is a different approach that combines the upgrade and Unicode conversion, which is called the TU&UC procedure.

On a copy of the production system, which is the *twin system*, you perform the SAP standard upgrade from the supported start release to ECC 6.0 non-Unicode (or to SAP R/3 Enterprise or ECC 5.0 non-Unicode), without any Unicode conversion-specific activities. Then you perform all of the MDMP-specific scans with transaction SPUMG, and you maintain the vocabulary and reprocess logs. You perform the ABAP Unicode enabling of customized programs and interfaces in the twin system with a dedicated process that outlines how ongoing changes and fixes for the production system will be handled until the go-live weekend.

In the meantime, you need to prepare the production system and upgrade to ECC 6.0 (or SAP R/3 Enterprise or ECC 5.0) using the standard upgrade tools PREPARE and SAPup, but you do the upgrade without any of the MDMP activities described earlier. Directly after the upgrade of the production system (still non-Unicode) is completed, the MDMP results (all scans, vocabulary, and reprocess logs) from the twin system are transferred to the production system very quickly. In summary, you copy the contents of all scan tables to the production system and then the

Uptime	Downtime
Prepare the SAP system for the upgrade using transaction SPUM4 and the PREPARE tool.	Perform the upgrade with the SAPup tool.
Upgrade a production system copy (sandbox) to ECC 6.0 and perform the Unicode enabling of customized programs and interfaces.	During the upgrade, perform several Unicode-specific activities, such as Unicode nametab generation.*
Create a transport with Unicode-enabled programs to bind them to subsequent upgrades.	After the upgrade is completed, perform several preparations for Unicode conversion.
Prepare for the upgrade using the PREPARE tool, including the import of the MDMP preparation tool SPUM4 in 4.6C.	Run the Unicode conversion of the database, including using suitable tuning tools.
Perform MDMP preparation in 4.6C using SPUM4.	Complete post-processing steps in ECC 6.0 Unicode starting with the SUMG automatic repair and others as described in the documentation.
Use the Downtime Minimized** strategy for upgrade.	

* SAP nametabs contain the runtime versions of active table and field definitions. Because of the different character lengths in non-Unicode and Unicode systems, the nametabs must be generated for the new Unicode system in a special way. See the documentation for more details.

* * SAP upgrades offer two strategies to influence the upgrade downtime and necessary hardware resources: The Minimized Downtime strategy allows a maximum of production time and a minimum of downtime during the upgrade and requires additional hardware. The Resource Minimized strategy stops the production system at the beginning of the upgrade and needs less additional hardware, but the upgrade downtime is longer.

Figure 12 Overview of the CU&UC procedure defined in terms of uptime and downtime

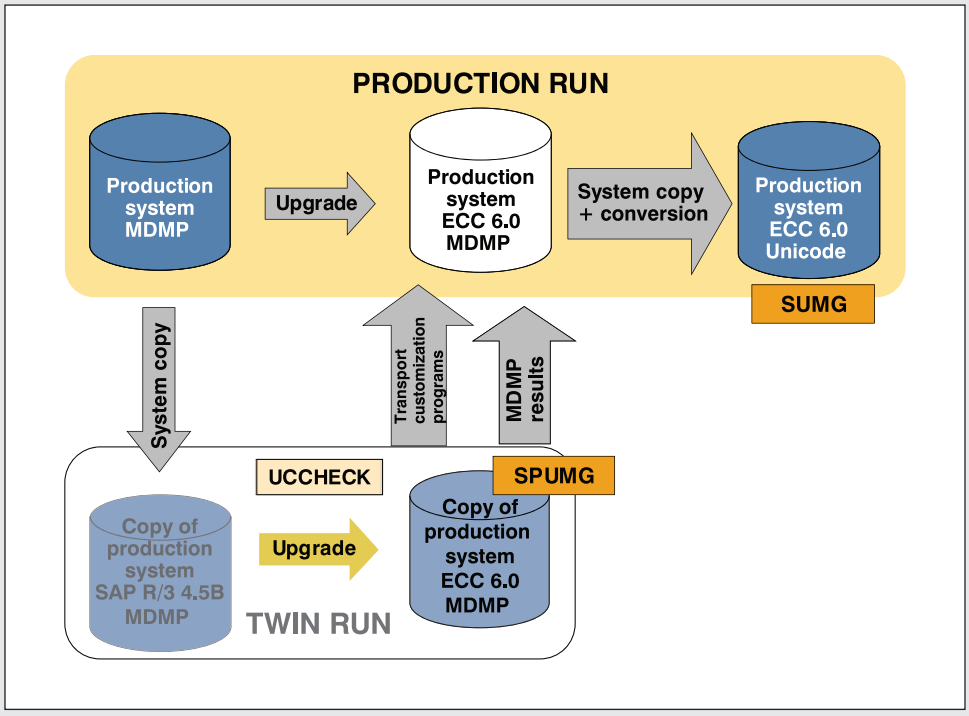

Figure 13 The TU&UC procedure

Production down/up	Production system	Twin system
Up	Finalize and freeze the production system.	Prepare the twin system hardware.
	Copy production to twin	
	Production is frozen; at a minimum, do not allow more changes to dictionary objects.	Upgrade the twin system to ECC 6.0 with the standard PREPARE and SAPup tools using the Resource Minimized strategy.
		Perform Unicode enabling of customized ABAP programs and interfaces by transporting them from other systems and doing delta work.
	Prepare the upgrade to ECC 6.0 with the PREPARE tool.	Run all SPUMG scans with the import of previously processed vocabulary, hints, and reprocess logs.
	Perform the upgrade to ECC 6.0 with the SAPup tool using the Downtime Minimized strategy.	
Down	Complete the upgrade post-processing steps.	Complete SPUMG.
	Transfer the SPUMG results from twin to production **Transfer the ABAP-enabled programs and interfaces from twin to production**	
	Perform delta activities.	
	Run the Unicode conversion of the database, including using suitable tuning tools.	
	Run SUMG automatic and manual repair activities.	
Up	Perform post-processing steps and make the system available in Unicode.	

Figure 14 Overview of the TU&UC procedure defined in terms of production uptime and downtime

ABAP Unicode-enabled customized programs and interfaces are transferred simultaneously as well. After this transfer, you can start the database export and import into the new Unicode system in the production system, and then perform repair activities using transaction SUMG. See **Figure 13** for an overview of the TU&UC procedure.

The twin system is used for collecting all necessary information required for a fast run in the production system and for creating the control information needed to correctly convert the MDMP data. The results are then reused during the conversion of the production system to minimize downtime.

This method is, in particular, suitable if:

- You have an MDMP system with a release earlier than 4.6C.
- You have an MDMP system with a very large database of several terabytes and cannot afford to run the MDMP database scans in the production system.
- You want a combined upgrade and Unicode conversion procedure for an upgrade to SAP R/3 Enterprise or SAP ECC 5.0 system.

Figure 14 summarizes the steps and system activities involved in the TU&UC procedure, classified by downtime and uptime.

Conclusion

To support a company doing global business, Unicode is the only solution for any global IT system

handling several languages and different character sets. SAP declared Unicode as its language support standard and made all SAP components Unicode-compatible as of SAP Web AS 6.20 and later. New SAP products will be shipped in Unicode only. The previous MDMP solution for multiple code page support in a single SAP R/3 system is no longer supported as of SAP ERP 6.0 and all other SAP Business Suite components based on SAP NetWeaver 7.0.

Remember that when considering a Unicode conversion, you need to take the following into consideration:

- New releases of SAP NetWeaver and SAP NetWeaver-based applications that are released in 2007 or later will only support new installations of Unicode systems.
- Since SAP Web Application Server (SAP Web AS) 6.20, all SAP applications are supported in Unicode, so you should convert non-Unicode SAP systems to Unicode as soon as possible, particularly systems that support blended code page or MDMP configurations.
- Because SAP NetWeaver 7.0 and SAP ERP 6.0 no longer support MDMP systems, SAP offers special procedures for a combined upgrade and Unicode conversion for software releases before SAP Web AS 6.20.
- SAP NetWeaver 7.0, SAP ERP 6.0, and other SAP Business Suite components still support SAP single code page systems; however, more and more IT vendors are shipping their solutions as Unicode-only, making single code page systems outdated. SAP recommends converting single code page systems to Unicode as well.

