Nils Bürckel, Alexander Davidenkoff, Detlef Werner

Unicode in SAP® Systems
Contents at a Glance

1 Introduction ............................................................ 13
2 Language Support in SAP Systems ......................... 27
3 Implementing Unicode in SAP Applications .......... 57
4 Guidelines for Unicode Projects ......................... 233
5 Languages and Translation ................................. 253
6 Summary ................................................................. 275
A Code Page Tables .................................................... 279
B Languages in an SAP Unicode System .................. 287
C Literature References and Other Support .............. 299
D Glossary ................................................................. 305
E The Authors ............................................................. 311
Contents

Preface .................................................................................................................. 11

1 Introduction ........................................................................................................ 13
  1.1 Globalization and Localization .......................................................... 13
  1.2 Language Support in IT Systems ...................................................... 16
  1.3 Unicode .............................................................................................. 19
    1.3.1 Character Encoding .......................................................... 20
    1.3.2 Java and Unicode .......................................................... 21
    1.3.3 Unicode Development .................................................. 22
    1.3.4 Modern Business Processes and New Markets ..................... 23
  1.4 Summary ............................................................................................. 25

2 Language Support in SAP Systems .......................................................... 27
  2.1 From Single Code Page Systems to Unicode .................................... 27
    2.1.1 Characters, Character Sets, and Code Pages ......................... 28
    2.1.2 Historical Development of Language Processing at SAP .......... 29
    2.1.3 Single Code Page Systems ................................................ 32
  2.2 Combination of Languages in a System ............................................. 36
    2.2.1 “Fool the System” .......................................................... 38
    2.2.2 MNLS System .............................................................. 38
    2.2.3 Blended Code Page System .............................................. 39
    2.2.4 MDSP System .............................................................. 40
    2.2.5 MDMP System .............................................................. 41
  2.3 Unicode in the SAP System ................................................................. 48
    2.3.1 Multilingual Data Exchange ................................................. 48
    2.3.2 Unicode Basics .............................................................. 49
    2.3.3 Unicode Formats in SAP Systems ........................................ 51
  2.4 Transition to SAP NetWeaver and Enterprise SOA with Unicode ...... 52
  2.5 Summary ............................................................................................. 55

3 Implementing Unicode in SAP Applications .............................................. 57
  3.1 Unicode Architecture ........................................................................... 57
  3.2 Unicode Conversion ............................................................................... 60
Contents

3.2.1 Generally Necessary Conversion Steps in a System .................................................... 61
3.2.2 MDMP and Blended Code Page-Specific Steps .................................................... 67
3.2.3 Export and Import ........................................................................................................... 95
3.2.4 SAP GUI for Windows ................................................................................................. 100
3.2.5 Printing in the Unicode System .................................................................................. 106
3.2.6 Transport between non-Unicode and Unicode Systems ........................................... 107
3.2.7 Conversion of SAP NetWeaver BI and SAP CRM Systems ......................................... 108
3.2.8 Summary ......................................................................................................................... 109

3.3 Upgrades and Unicode Conversion .................................................................................... 109
3.3.1 Sequential (Separate) Upgrade and Unicode Conversion ............................................. 113
3.3.2 Combined Upgrade and Unicode Conversion .............................................................. 113
3.3.3 Twin Upgrade and Unicode Conversion ....................................................................... 123
3.3.4 Comparison of the Combined Procedures .................................................................... 126

3.4 ABAP and Unicode ............................................................................................................. 128
3.4.1 Overview ......................................................................................................................... 129
3.4.2 Unicode-Relevant Changes and Additions in ABAP as of SAP Web AS 6.10/6.20 ........... 131
3.4.3 Tools for Unicode Enabling .......................................................................................... 147
3.4.4 Summary ......................................................................................................................... 154

3.5 Communication and Interfaces .......................................................................................... 154
3.5.1 Homogeneous and Inhomogeneous Communication ..................................................... 156
3.5.2 Communication with RFC ............................................................................................ 160
3.5.3 Communication Using File Transfer .......................................................................... 173
3.5.4 Communication Between SAP Unicode and SAP non-Unicode MDMP Systems ........ 178

3.6 Expansion into New Countries with Unicode .................................................................... 217
3.6.1 Technical Configuration of a New Language in SAP Unicode ..................................... 218
3.6.2 Case Study: Introduction of Vietnamese in an SAP Unicode System ......................... 221
3.6.3 Summary ......................................................................................................................... 228

4 Guidelines for Unicode Projects ........................................................................................... 233
4.1 New Installation .................................................................................................................. 233
4.2 Unicode Conversion .......................................................................................................... 234
Contents

4.2.1 Typical Steps in a Conversion Project for an Environment with Three Systems ................... 234
4.2.2 Determining Factors of a Conversion Project .. 241
4.2.3 Comparison with Upgrade Projects and OS/DB Migration ................................. 244
4.2.4 Converting Complex Landscapes ..................... 247
4.3 Release Changes and Unicode Conversion .......... 248
4.4 System Consolidation ................................................. 251
4.5 Summary ................................................................. 252

5 Languages and Translation ................................. 253
5.1 Language and Translation Management .......... 254
5.1.1 Translation Objects .......................................... 255
5.1.2 Language Installation ...................................... 256
5.2 Customer-Specific Translations ...................... 257
5.2.1 Translation Strategies ..................................... 257
5.2.2 SAP Translation Tools ..................................... 259
5.2.3 SAP Translation Workbench ......................... 261
5.2.4 Language Transport ........................................... 262
5.2.5 Language-Dependent Customizing .......... 266
5.2.6 Address Versions ............................................ 271
5.3 Summary ................................................................. 273

6 Summary ................................................................. 275

Appendix ................................................................. 277
A Code Page Tables .................................................. 279
B Languages in an SAP Unicode System .............. 287
C Literature References and Other Support .......... 299
C.1 Literature ............................................................. 299
C.2 Links ................................................................. 300
C.3 SAP Notes ............................................................ 301
C.4 Documents Needed for Unicode Conversions ...... 303
C.5 Typical Process of Testing a Unicode Conversion .... 303
D Glossary ................................................................. 305
E The Authors .......................................................... 311

Index ................................................................. 313
Preface

The only constant is change. Not only are there a daunting number of languages, divided into language families, but every language inevitably changes over even a relatively short time. Continuous communication is the reason.

A significant challenge during the fast-paced development of information technology and computers was therefore the attempt to encode language and the characters associated with it into a form suitable for machines, so as to be able to store and exchange data. Data exchange was, and still is, particularly challenging, since one must define certain standards in order to ensure the smoothest possible data exchange between different computers and programs. Over time, it became clear that the variety of different formats introduced—particularly with increasing globalization—were still unable to represent languages sufficiently well, and that there were even errors during data exchange between heterogeneous IT platforms.

The solution to this omnipresent problem was the introduction of Unicode. For the first time, a globally accepted, uniform standard had been created that, thanks to the fixed assignment of one number to every character, guarantees that texts in any language can be displayed and transmitted without error, both today and in the future.

“When the world wants to talk, it speaks Unicode.”\(^1\) Besides the basic principles of Unicode, this book also describes how Unicode is supported by SAP as the future technological basis for all its software products. This book is intended for a broad audience: from the top manager or CIO who needs to decide how the enterprise’s ERP system can best be converted to use Unicode, to the project manager who introduces new Unicode-based software, to the IT specialist who needs to know the steps to make in-house software Unicode-capable or how an existing ERP system can be converted.

\(^1\) This was the motto of the 20th International Unicode Conference in 2002.
Structure  After a general introduction in Chapter 1, you will read in Chapter 2 how languages are generally supported in SAP systems. Chapter 3 then covers Unicode-based installations and shows how existing solutions can be converted. It first describes the basic architecture of an SAP Unicode system (Section 3.1). Based on that foundation, there is a detailed description of a Unicode conversion of an MDMP system (Section 3.2). In Section 3.3, we will discuss combining an upgrade with a Unicode conversion. ABAP enabling and the handling of interfaces form a very large part of a Unicode conversion, so Sections 3.4 and 3.5 are devoted to these topics separately. To conclude the chapter, we will describe the options for expansion into countries for which Unicode is a requirement (see Section 3.6). How you can effectively manage these projects is the topic of Chapter 4, after which Chapter 5 discusses the topic of translation. Chapter 6 summarizes the key points of the first five chapters with respect to Unicode installation and conversion.

The book is structured so that you can read selected chapters or sections independently. For instance, a programmer with basic Unicode knowledge could simply read the information in Section 3.4 regarding "ABAP and Unicode" to discover how programs can be verified for execution in a Unicode system and, if necessary, how they must be changed.

In all explanations, great value is placed on specific practical examples that we, the team of authors (from SAP Globalization Services), can guarantee. We all have years of experience with the introduction of Unicode into SAP software, whether in internal SAP projects like the Unicode Enabling of SAP R/3 Enterprise or the global support of SAP customers, as with Unicode conversions.

Acknowledgements  The authors would like to take this opportunity to thank all the SAP customers, partners, and colleagues who helped with the creation of this book, particularly for all the valuable discussions. Our particular thanks go to the management of SAP Globalization Services for the opportunity to write this book.

Nils Bürcikel, Dr. Alexander Davidenkoff, Dr. Detlef Werner
Walldorf, St. Leon-Rot, February 2007
All SAP applications are available based on Unicode. In this chapter, you will learn how you can optimally plan and execute a new installation or the conversion of an existing system.

4 Guidelines for Unicode Projects

Today, all SAP applications are available in Unicode-based versions, and new software products from SAP such as the SAP NetWeaver Exchange Infrastructure (SAP XI) or the SAP NetWeaver Portal are now only delivered as Unicode versions. The support for obsolete solutions for the combination of languages and code pages—such as MDMP in SAP R/3—is being terminated step by step, so that SAP ERP 2005 no longer supports MDMP. As of 2007, all new installations of applications based on SAP NetWeaver will only be possible under Unicode.

4.1 New Installation

The new installation of a Unicode-based system is basically no different from that of a non-Unicode installation. However, the following should be taken into consideration in the implementation project:

- As mentioned in Chapter 3, a Unicode system has somewhat higher hardware requirements; these must be taken into account.
- If there are interfaces to non-UniCode systems, the configuration must be undertaken with care, in order to eliminate such risks as data loss because of incorrect conversion.
- The same applies for uploads or downloads of files which come to the system over the network (by means of FTP, for example).
- If third-party software will be integrated, you must carefully check that it will function smoothly with the SAP Unicode installation. A request to the appropriate SAP software partner will usually suffice.
When creating custom developments, you should proceed in compliance with Unicode, as described in the SAP Unicode Enabling Cookbook (see Appendix C) and in Section 3.4.

Tip
SAP Globalization Services offers different services and workshops that explore individual Unicode aspects in detail according to customer requirements, either before or during a project. The SAP Unicode Workshop is the ideal introduction for all Unicode projects. You can find a list of all services at SAP Service Marketplace (http://service.sap.com/globalization) in the Service Offerings area.

4.2 Unicode Conversion

4.2.1 Typical Steps in a Conversion Project for an Environment with Three Systems

You can see a rough overview of the conversion of one SAP system in Figure 4.1, which shows the typical phases of a conversion. The plan for a conversion is always the same: Preparation takes on a very important role, followed by the conversion itself, and then the phase of postprocessing.

In the following sections, we will describe in more detail how a Unicode conversion project for a three-system environment might be structured. In the evaluation phase, we work out the consequences Unicode will have for the existing system environment and which Unicode strategy is to be followed. The actual preparation phase for a particular system environment follows. Then, a sandbox conver-
sion is carried out as the first test environment. The sandbox conversion may need to be repeated several times. In parallel with this, the development system (DEV) or the consolidation system (QAS) can already be converted. After all the tests in QAS and in the sandbox system have been completed successfully, the conversion of the production system (PRD) follows.

The steps presented here must be adapted to the actual customer situation. This list also makes no claim to completeness.

**Project Evaluation Phase**

Before any Unicode conversion project, we must clarify what Unicode means in the specific situation of the client. In general, cost and effort are the focus. However, we also should consider what efforts will be caused and what consequences will result if Unicode is not introduced. It is often necessary to have a “business case” to justify extensive investments in new technology.

At the end of this phase, the general Unicode strategy of the company regarding SAP technology should be clear. However, the strategy may still need to be adapted as conditions change, so its validity must be verified at regular intervals.

The following points describe possible factors to be taken into account in this step:

- **Unicode conversion process and consequences**
  In this phase, it is important to understand the principle of Unicode conversion. What steps and effort are necessary, and where must the customer make adaptations?

- **Acquiring relevant customer-specific information**
  To be able to produce a reasonable Unicode strategy, the information on Unicode available from SAP must be complemented with the customer’s own data. The following list includes the most important points:
  - Overview of the system landscape (systems, releases, support packages, front-end software, and so on)
  - Database sizes (in GB), the 50 largest tables, and the hardware configuration of all relevant systems
Requirements pertaining to tolerable downtime for individual systems

- Code page setup of all systems (Unicode, MDMP, single code page, blended code page)
- Description of the interfaces between the systems and to non-SAP systems (“rough interface catalog”)
- Existing add-on solutions (SAP and non-SAP)
- Number and type of existing custom developments
- Existing rollout plans in other countries for the different systems
- Planned system mergers
- Possible conversion strategies

Attention to these points generally results in the conversion sequence for the existing systems. For the systems to be converted, the first option at this point is to discuss the procedures for conversion strategies (for instance, the creation of a sandbox system) and also ways to minimize downtime (see Chapter 3).

- **Evaluation of the consequences should Unicode conversion be postponed**

Customers should consider carefully what the postponement of Unicode means from short-term and long-term perspectives. SAP Note 79991 regarding the SAP support for non-Unicode should be considered carefully in this context (see Table 2.7 in Chapter 2).

- **Experience of other customers**

An exchange of experiences with other customers at the start of the project and also over the course of the project promotes better understanding of possible problems and workarounds.

- **First “rough” estimate of effort**

A first approximate estimate of effort should consider the points of hardware requirements, resource requirements, and project duration.

- **Business case and creation of initial project plan**

The analysis above allows a preliminary project plan to be created. This plan generally doesn’t include any exact details of the conversion.
Project Preparation Phase

In the preparation phase of the project, we begin the system-specific preparations for the coming Unicode conversion. Conditions and limitations must be checked and patches may need to be installed. The necessary hardware must be procured. All available documents on Unicode conversion should also be studied. It is recommended to create a detailed project plan for the coming sandbox conversion. The following points should be taken into consideration:

- **Detailed prerequisites and restrictions**
  If products from other manufacturers are used in the SAP environment, we recommend that the Unicode capability of these solutions should be certified by the manufacturers. In general, you can’t assume that an SAP certification also means Unicode capability.
  For relatively "old" patch levels in SAP solutions, there is a high probability that an update will be necessary. For extensive changes (SAP GUI for Windows, for example), the tests should first be performed with the current patch level before a global rollout is started.

- **Interface catalog with language dependencies**
  A list of all interfaces and their language dependencies (e.g., "are tests with special characters transmitted?") helps identify critical areas.

- **Hardware procurement**
  For larger systems, a sandbox conversion is absolutely recommended. The temporary hardware needed for this must be provided. If the hardware is to be updated during the production conversion, this must be planned in advance.

- **Options for performance optimization**
  To prepare a sandbox conversion, we should have examined what means are available for runtime optimization and what will make sense for the initial conversion. It may well be that certain methods are not yet fully available. In that case, additional steps may be necessary.

- **Determination of the strategy for handling ABAP objects (UCCHECK)**
  In this example, we will assume that the ABAP objects are already adapted in advance (see the subsequent bullet point on ABAP).
Guidelines for Unicode Projects

Archiving before Unicode conversion

▷ Use of all possible archiving options to reduce database size
By reducing the size of the database, it is possible to improve several aspects of the conversion. These include the export and import times for the production conversion. The runtime of Transaction SPUMG will be shorter if there is less data to scan.

▷ Necessary documentation and SAP Notes
All necessary guidelines and associated SAP Notes should be downloaded. During the course of the project, we recommend that you check regularly whether there are changes to the SAP Notes or in the conversion guidelines.

▷ Enabling of ABAP objects (Unicode enabling)
The handling of ABAP objects is fully possible only as of Basis Release 6.10 using Transaction UCHECK. If the release is 6.10 or higher, the objects can be adapted in advance. The objects are then ideally executable either under under Unicode or in a non-Unicode environment. Since the result is a certain independence from the Unicode conversion, the UCHECK processing is performed as a separate task. For new developments (creation of new ABAP objects) the Unicode flag is set by default as of SAP Web AS 6.10, so that in this case Unicode capability is largely assured. Existing objects without the Unicode flag that must be adapted during ongoing maintenance should also be made Unicode-capable now, because these objects then must only be tested once.

Structure and Conversion of the Sandbox System

“Proof of concept”

Once the prerequisites for the sandbox conversion have been fulfilled, you can carry out the first test conversion. Care should be taken that even during the first attempt the known performance optimizations are used during export and import. The following list provides an overview of the steps:

▷ Structure of the sandbox system as a copy of PRD
In the ideal case, the hardware of the sandbox system already corresponds roughly to the hardware of the PRD system for this initial test. In that case, the results of the scan times and the export/import times can be accurately used in estimating the behavior of the production system.

In practice, however, there is usually no comparable hardware available, so that at first the process of conversion itself is being
tested with no real focus on the runtimes. For large systems, however, this can result in very long runtimes, so that the entire test takes a very long time.

- **Preparatory steps for Unicode conversion**
  The main area for preparation for Unicode conversion of an MDMP system is Transaction SP UMG. Sufficient time must be budgeted for the scans which build the system vocabulary, for processing of the vocabulary, and also for the reprocessing scans. The runtimes of the scans for large systems run in the range from a few days to several weeks. The scan without language codes generally requires the longest time, but the reprocessing scan can also require a large percentage of the project time.

  Time also is needed for assignment of the vocabulary entries. Because the system vocabulary maintained in the sandbox system forms the basis for the later conversion of all existing systems, this process should be executed immediately at the outset with all necessary care.

- **Export and import**
  This area includes the tests relevant to runtime optimization. If Transaction SPUMG has already been sufficiently well tested, it is not necessary to execute this transaction repeatedly as a preparation scan. This would take—in the worst case—several weeks each time. Alternatively, a backup after completion of the SPUMG work can be used as a starting system.

- **Execution of post-conversion tasks**
  Transaction SUMG covers the most important reprocessing steps. This transaction can be used to estimate the effort for manual repair. Repair hints may already be generated here.

- **Test phase**
  After the Unicode conversion is complete, tests of the applications follow. For instance, a part of the custom ABAP objects can be tested. If possible, interface tests should also be performed at this point. Normally, however, this is only fully possible in the QAS, because in the sandbox system the interfaces must first be set up.

  However, it is always the case that all tests that can already be performed in the sandbox system should be performed there. The earlier a problem is discovered, the more time you will have to correct it.
Analysis and results of the sandbox conversion

The sandbox conversion provides results for the R3load runtime optimization, the procedure, and the scan runtime for Transactions SPUMG and SUMG, as well as for initial Unicode testing after the conversion phase. Based on these results, you now must decide whether additional sandbox conversion runs are needed and when the next systems in the landscape should be converted. There follows a corresponding refinement and adaptation of the project plan to the continuing procedure.

Repetition of the sandbox conversion (depending on the results of the previous conversion)

The sandbox conversion should be repeated until the requirements for downtime are met. In parallel with this, the conversion of the QAS or the DEV system can be started.

Converting the Development or the Quality Assurance System

The conversion of the DEV or the QAS system is performed analogously to conversion the sandbox system. Under some circumstances, it may be possible to accept longer downtimes here than with the production system.

Meaningful interface and integration tests are generally only possible on the quality assurance system. A parallel preliminary test of Transaction SPUMG on the PRD system at this point will ensure that the scan duration of this transaction is not underestimated.

Converting the Production System

Based on the experience with the sandbox system as well as the DEV and QAS systems, the conversion of the production system can now proceed. It is important that the conversion is performed under exactly the same conditions as those pertaining to the test system. For instance, the patch level of the kernel (particularly the R3load version) should be identical to that of the other systems.

The first day or the first week after the conversion generally requires particular attention, since there may be problems in some areas. You should take this into consideration in your planning.
4.2.2 Determining Factors of a Conversion Project

At the outset of a Unicode project, there is a typical series of questions, which will be clarified in this section. The following list shows initial possible reasons for a Unicode conversion:

- The organization desires an upgrade of the existing MDMP system to SAP ERP 2005 (see SAP Note 79991).
- English should be possible as the central logon language for all countries or languages.
- The data exchange between MDMP and Unicode causes problems.
- Java technologies (such as ESS/MSS on SAP ERP 2004) should be used in the MDMP environment.
- Rollout is needed in other countries that are not covered by the existing non-Unicode solution in the system.
- The organization needs to consolidate systems with different code-page configurations.
- The organization anticipates rollout in countries whose characters are not supported except in Unicode (for example, Arabic or Vietnamese).
- Support of dialects (such as Canadian French) is needed.
- The organization needs to display certain characters (e.g., the € character) that are not supported except in Unicode.
- Internet connection (e.g., a Web store) is needed.
- Strong Java integration of the system is needed.

The project duration for a Unicode conversion, like that for an upgrade, is determined by many factors. The main areas are shown in Table 4.1. In addition, the duration depends on the number of resources available and their state of knowledge.

<table>
<thead>
<tr>
<th>Effort</th>
<th>Factors</th>
<th>Easy</th>
<th>Medium</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language technology used</td>
<td>Single code page</td>
<td>Asian code page</td>
<td>MDMP</td>
<td></td>
</tr>
<tr>
<td>SAP solution used</td>
<td>SAP Web AS (standalone)</td>
<td>SAP ERP or SAP R/3 Enterprise</td>
<td>SAP CRM with Mobile Sales</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1 Effort Involved in a Unicode Conversion
As a minimum value for the conversion of a three-system landscape, you can assume about four weeks of project runtime. On average, these projects take about three to four months. For very large MDMP systems with very many custom ABAP objects or interfaces to other MDMP systems, the runtime can even be more than a year.

A cost estimate, too, depends on the factors listed in Table 4.1. Moreover, in this area the additional hardware requirements must be taken into account. Expenses for testing and for the adaptation of custom reports may under some circumstances represent a large part of the overall budget.

A Unicode project also requires expertise in the area of ABAP enabling as well as in the interface area. Specialists in the programming environment and employees with SAP NetWeaver knowledge will be needed. Transaction SPUMG is generally executed by SAP NetWeaver experts. For the preparation of the system vocabulary, however, people will also be needed who know each language. The export and import procedures and optimization are comparable to an upgrade and also require technical knowledge. Testing is generally the responsibility of the application.

<table>
<thead>
<tr>
<th>Factors</th>
<th>Easy</th>
<th>Medium</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database size</td>
<td>Database &lt; 300 GB</td>
<td>Database between 300 GB and 1,500 GB</td>
<td>Database &gt; 1,500 GB</td>
</tr>
<tr>
<td>Downtime accepted</td>
<td>Downtime &gt; 4 days</td>
<td>Downtime from 2 to 4 days</td>
<td>Downtime &lt; 2 days</td>
</tr>
<tr>
<td>Hardware properties</td>
<td>Very fast hardware</td>
<td>Medium hardware</td>
<td>Slow hardware</td>
</tr>
<tr>
<td>ABAP objects enabling</td>
<td>Small number of objects</td>
<td>Medium number of objects</td>
<td>Large number of objects</td>
</tr>
<tr>
<td>Conversion method</td>
<td>Standard</td>
<td>Split of large tables</td>
<td>IMIG/CU &amp; UC</td>
</tr>
<tr>
<td>SAP interfaces</td>
<td>Unicode systems</td>
<td>Single code page systems</td>
<td>MDMP systems</td>
</tr>
<tr>
<td>Non-SAP interfaces</td>
<td>Unicode systems</td>
<td>Single code page Latin-1</td>
<td>Asian code page</td>
</tr>
</tbody>
</table>

Table 4.1 Effort Involved in a Unicode Conversion (cont.)
Figure 4.2 shows two possible scenarios for a Unicode conversion: On the left is a conversion with a sandbox, and on the right you see a scenario without a sandbox, where the QAS is converted before the development system.

The conversion using a sandbox is the procedure recommended by SAP for a Unicode conversion. However, this still doesn’t answer the question of when a sandbox conversion is absolutely necessary. In the following cases, a sandbox conversion is difficult to avoid:

- **Critical requirements for downtime or conversion of very large systems**
  Conversion runtimes can only be meaningfully tested on sandbox systems with comparable hardware parameters. Extrapolation of runtimes from very weak servers to production environments is extremely uncertain. As a result, the probability of surprises during the production conversion increases if sufficient testing does not take place under comparable conditions.
Guidelines for Unicode Projects

- **MDMP conversion**
  In the case of an MDMP conversion, SAP strongly recommends a test conversion of a copy of the production system. This is the only way to completely test the conversion.

- **Combined upgrade and conversion solutions**
  The basic prerequisite for more complex solutions like CU & UC (see Section 3.3) is the construction of separate test systems. In the case of TU & UC, a twin system must be built anyway.

- **QAS conversion before DEV conversion**
  It is worth considering whether the QAS should be converted before the development system. A "proper" test of the interfaces is often only possible on the QAS. Problems with interfaces, however, should be identified quickly, given that correction may be difficult under some circumstances. In principle, the same thing applies for a meaningful integration test.

- **Conversion with "Maintenance landscape"**
  Analogous to upgrades, there is a scenario of non-Unicode maintenance systems (DEV and QAS) which will support the non-Unicode production system during the Unicode conversion. This ensures that a critical problem that occurs differently in a non-Unicode environment than with Unicode can be corrected quickly.

### 4.2.3 Comparison with Upgrade Projects and OS/DB Migration

On the topic of upgrades, there is a great deal of documentation, and most customers already have been through at least one upgrade project. Some customers also have experience in the area of OS/DB migration. With this type of background, the following comparison of a Unicode conversion with an upgrade and OS/DB migration can be quite helpful:

- **Functionality**
  During an upgrade, applications may change thoroughly, so that existing custom development may need to be adapted. In a Unicode conversion or OS/DB migration, the application end of the system changes only minimally.
- **System Downtime**
  The dependency of downtime for the conversion of the production system on the database size is significantly larger for a Unicode conversion. In an upgrade, this dependency is generally compensated for by hardware, which tends to be better on a larger system. This trend can also be seen in Unicode conversions, but corresponding efforts must be made to optimize performance.

  The downtime in a Unicode conversion is comparable to the downtime in an OS/DB migration. However, the CPU requirements for a Unicode conversion are significantly higher, because the compressed cluster tables, for instance, must be subjected to many computationally intensive operations (the data is decompressed, converted, and then recompressed).

- **Comparison with UCCHECK and SPUMG Enabling**
  The adaptations in the area of custom development (through Transactions SPAU or SPDD) can best be compared to the UCCHECK enabling; however Transaction UCCHECK shows directly where objects must be changed. Moreover, the objects can be adapted in the non-Unicode system, even before Unicode conversion. This generally doesn’t apply for an upgrade. For an MDMP system, Transaction SPUMG corresponds to no particular step in an upgrade. A scan of the entire database and the associated maintenance of the results are not necessary for an upgrade.

  Adaptations in the interfaces for a Unicode conversion, particularly for an MDMP system, require particular attention (see Chapter 3). In the case of an OS/DB migration, adaptations must be made to custom ABAP objects which contain database-specific commands (e.g., Native SQL).

- **Release Limitations**
  In an upgrade, the release of the SAP system changes after the process is finished. Generally, an upgrade is possible in the source system for most of the releases still supported.

  A Unicode conversion, on the other hand, is executed by default on a single release. There are limitations, both on the Unicode side (R/3 releases up to 4.6C don’t support Unicode) and on the non-Unicode side (MDMP or blended code pages are no longer supported as of SAP ERP 2005).
In an OS/DB migration, there are only a very few release limitations. Generally, a migration can be performed on any supported release. Just as for a Unicode conversion, the release does not change during a migration.

- **Development Freeze During the Project**
  In an upgrade, transportation between the different releases is not supported. That means that a development freeze during the upgrade project can hardly be avoided.

  In the case of a Unicode conversion, it is quite possible to perform a transport between Unicode and non-Unicode releases (with limitations where MDMP is concerned). That means that a development freeze may not be absolutely necessary. SAP’s recommendation, however, is to minimize the number of projects as far as possible during the Unicode conversion, as there may very well be differences between Unicode and non-Unicode systems (for instance, where interfaces are concerned).

- **Frequency**
  A Unicode conversion must be performed exactly once per system. After that, no more projects are necessary. An upgrade, on the other hand, must generally be performed on a regular basis. An OS/DB migration lies somewhere in between. For most customers it is a rather unusual task and is not repeated often for each system.

- **End User Training**
  For an upgrade, end users must often be trained, since the functionality may have changed.

  For a Unicode conversion, there are only a very few areas where the users even notice that there has been a technical change. Examples are the Word editor for SAPscript editing and uploads and downloads, where a target code page may need to be given under some circumstances. If the Unicode conversion is combined with an upgrade, however, there must be corresponding training for the end users.
4.2.4 Converting Complex Landscapes

Besides an ERP application, many customers use a number of other SAP applications. These are generally connected, and the question immediately arises as to the order in which the different systems should be converted.

It is generally impossible to convert all systems in one weekend, because there would be an extremely great effort associated with that. But the risk that the entire conversion fails on one weekend is also very high for such a “big bang” conversion. If you assume that five closely connected production systems are to be converted in a single weekend, then no system can be allowed to have a significant program that cannot be corrected immediately, because otherwise all the systems will need to be reset.

Based on the following criteria, a customer-specific evaluation should be made as to the optimum order:

- **Unicode capability of the different SAP applications (releases)**
  The prerequisite for Unicode conversion is the Unicode capability of the release. For instance, if a system is running on SAP R/3 4.6C, the conversion can only take place after an upgrade.

- **Requirements pertaining to languages and countries in the different systems**
  Systems that must handle data from multiple “old code pages” should be higher-priority candidates for Unicode conversion. Systems without corresponding requirements (purely single code-page systems) can be converted at a later date.

- **Minimization of interfaces between MDMP and Unicode systems**
  The interface problems described in Section 3.5 between Unicode and MDMP systems leads to the very important criterion that the number of MDMP systems in the system environment should be kept as low as possible.

- **Upgrade planning for MDMP systems based on SAP NetWeaver 2004s**
  As of SAP NetWeaver 2004s, MDMP is no longer supported (see SAP Note 79991). Thus, for upgrade planning for corresponding solutions, Unicode conversion should be planned as well.
Consolidation plans
For planned consolidations, particularly when different code pages are in use, a conversion to Unicode is a good alternative.

Database sizes of different systems and quickly growing databases
The effort of a Unicode conversion depends strongly on the size of the database. If there are systems in the environment that have a high rate of growth, then you should consider converting that system earlier rather than later.

Planned projects and regular actions
Unicode conversion should ideally happen in a “project-sparse” time. Ideally, existing projects should be postponed until after the conversion. But critical projects like annual reporting must also be taken into account. Unicode conversion shortly before annual reporting is generally problematic.

4.3 Release Changes and Unicode Conversion
Upgrades (release change) and Unicode conversions are both projects in the course of which a great deal of application testing is necessary. Although these are two logically independent steps (see Figure 3.3 in Chapter 3) there is still the question of how well the two tasks can be combined.

In Section 3.3, appropriate combined procedures were already described. These are particularly interesting in relation to an upgrade from a non-Unicode-capable release with MDMP to SAP ERP 2005, because in that target release MDMP is no longer supported (see SAP Note 79991).

In summary, these are the possibilities for integration tasks into an upgrade:

- Separate projects
  The upgrade is executed completely separately from Unicode conversion.

- Consideration of ABAP objects during the upgrade
  The enabling of ABAP objects to the stricter Unicode syntax rules is considered during the upgrade. Separate integration tests for the changes to ABAP objects in non-Unicode are not necessary.
- **Upgrade and Unicode conversion in one project, but on separate weekends**
  It is possible to perform the conversion and the upgrade in one project, but to select a different weekend for the conversion of the production system. Assuming that the upgrade is performed before the conversion, this means that tests must be performed both in the non-Unicode and the Unicode systems, as in this case the non-Unicode system will be going live on the new release.
  The advantages of this approach would be that a sandbox system could be used both for the upgrade and for the conversion, and that tests may be performed twice, but otherwise would still be performed shortly after one another in an identical procedure.

- **Upgrade and Unicode conversion on the same weekend**
  The main criterion for the decision whether the conversion and the upgrade can be performed on one weekend is the runtime of both procedures. If a Unicode conversion already results in 30 to 40 hours of downtime despite the use of all possible optimization methods, a combined procedure with all the necessary preparation and postprocessing within 48 hours will be possible only in very occasional cases. As a result, either a longer downtime must be accepted, or the alternatives described in the previous points must be selected.
  A combined solution also means a significant increase in the complexity of the project. The procedure for an upgrade alone is well documented in great detail, and there is also documentation of separate Unicode conversion. The integration of the two projects requires detailed planning for which there is currently very little background information available. An example might be the handling of ABAP objects during an upgrade, with conversion starting from release SAP R/3 4.6C.

- **Combined Upgrade & Unicode Conversion**
  This CU & UC method was primarily developed for MDMP customers who have SAP R/3 4.6C as their source release and are working towards a technical upgrade to SAP ERP 2005. In SAP Note 928729, the releases supported and their limitations are described in more detail.
  A significant component of this technology is the use of Transaction SPUM4, which represents an equivalent to Transaction
SPUMG under SAP R/3 4.6C (see Section 3.3.2). This is necessary because the principle behind Transactions SPUMG and SPUM4 is that the transaction will be performed online during production operation. Because the runtimes of Transaction SPUMG for MDMP customers will run at least for a matter of days, the performance of SPUMG in the target release is impossible during the downtime. Thus SPUMG was implemented in SAP R/3 4.6C as SPUM4, so that online performance would be possible under this release.

In contrast, Unicode enabling (Transaction UCCHECK, for instance) is not possible under SAP R/3 4.6C. This preparation step must be performed on a sandbox system on a Unicode-capable release (on non-Unicode or Unicode). The results can then be transported into the production system after the upgrade is complete.

Figure 4.3 shows a schematic diagram of one possible procedure for a CU & UC. In the sandbox system (SBX), the Unicode enabling must first be done, and the results can then be used in the PRD system after the upgrade is complete. The upgrade and Unicode conversion in the sandbox system can alternatively be performed separately in the initial step so that the UCCHECK activity can proceed on the non-Unicode system.

![Figure 4.3 Sample Procedure for CU & UC](image-url)
> **Twin Upgrade & Unicode Conversion**

CU & UC cannot be performed for source system releases prior to SAP R/3 4.6C. In order to provide a solution for these customers, the method TU & UC has been developed. In this procedure, a parallel twin system is constructed as a copy of the production system, and an upgrade is performed without Unicode conversion (see Figure 4.4).

![Diagram of Twin Upgrade & Unicode Conversion](image)

**Figure 4.4** Sample Procedure for TU & UC

Transaction SPUMG can now be run on the copy. The results are then transferred to the production system for a production upgrade before the Unicode conversion and used there. Here, there is a delta (table entries that have entered PRD after the construction of the twin system), which becomes larger the older the system copy of the production system. As a result, you must take into account the fact that the tasks after conversion (SUMG) will take more time.

In SAP Note 959698, the releases supported and the limitations of TU & UC are described in more detail.

### 4.4 System Consolidation

For the purpose of simplifying the system landscape, there are efforts on the part of many customers to reduce the number of existing systems. One option is to integrate systems with similar applications but which have been constructed for other countries into one...
existing system. Here there is an opportunity to use a Unicode system as a basis.

In general, the systems to be migrated are non-Unicode systems, and conversion of the data during the migration would be desirable and usually possible. MDMP systems are again an exception in this respect, as the conversion is not as simple as for a single code page system. The SPUMG logic can currently not be used for a data migration.

Client migration server

SAP System Landscape Optimization has plans for tools which will make it possible to migrate from SAP R/3 4.6C (non-Unicode) to SAP ERP 2005 (Unicode). So here, three steps—an upgrade, a Unicode conversion, and a system consolidation—will be combined into one. The technology used will be called the Client Migration Server, or CMIS. For more complete and up-to-date information, you can consult the SAP Service Marketplace (http://service.sap.com/slo).

4.5 Summary

The focus of this chapter has been the planning for Unicode conversion in a complex system landscape. A new installation is relatively simple, because it hardly differs from the installation of a non-Unicode system. In the conversion of a three-system landscape, on the other hand, there are already many different options for the implementation of Unicode.

An estimate of the effort of the Unicode project is not possible without exact knowledge of a variety of factors. Here, database size, the possible use of MDMP, the number of custom programs, and the type and number of interfaces all play significant parts. A comparison of Unicode conversion with an upgrade project made it clear that, depending on the conditions, the Unicode conversion may easily take as much time as a release upgrade. A combined upgrade and conversion procedure is available for customers, but the complexity of such a project should not be underestimated.
Index

# Replacement character 172
1:n relation 272
7-bit ASCII 16

A

ABAP 128
ABAP Editor 129
ABAP enabling 114
ABAP programming 134
ABAP stack 160, 167
ABAP text pool entries 264
ABAP/Unicode check 148
Address versions 271, 272
Adobe Document Services 221
ALE 136, 305
ALE communication scenario 202
ALE distribution model 187
ALE inbound processing 188
ALE message type 187, 193, 202
Multilingual 202
ALE prototype 187
ALE Unicode scenario 191, 193, 202
Alignment gaps 134
Ambiguous blended code page 40, 70
ANSI 305
API 305
Append structure 115, 140
Application server 175, 305
Archiving 238
ASCII 162
German 306
Asian Double-Byte languages 131, 161
Asian languages 157
Asynchronous transmission 305

B

Backup 305
BAdI 149
Basic IDoc type 193
BCP → Blended code page 305
BDC 305
BDocs 180, 182, 204

BIDI technology 30, 305
Big Endian 51, 305
Big Five 35
Binary file 305
Bit option 172
Black box 136
Blended code page 109, 111, 112
Blended code page systems 39
BOM 305
BRIC countries 110
BTF 305
Business address services 272
Business Server Pages 54
Byte 305
Byte-swapped 305

C

Cascading Font Generator 228
Cascading fonts 107, 227, 228
Case 305
Lower 305
Upper 305
CBL 305
CGI 305
Character 28, 132
Asian 146
Character composition 34
Character elements 28
Character encoding form 305
Character encoding scheme 306
Character rendering 218
Character set 28
ASCII 162
Global 32
Vietnamese 221
Character-based data types 163
Charlen 161
CJK 306
CJKV 306
CL_ABAP_CHAR_UTILITIES 144
CL_ABAP_CONTAINER_UTILITIES 146
CL_ABAP_CONV_IN_CE 146
CL_ABAP_CONV_OUT_CE 146
CL_ABAP_CONV_X2X_CE 146
Index

CL_ABAP_LIST_UTILITIES 146
CL-NLS_STRUCT_CONTAINER 138
CL-NLS_STRUCT_CONTAINER_OFFS 139
CL-NLS_STRUCT_CONTAINER_SNAME 138
Client migration server 252
CMIS 252
CMOD 214
Code Inspector 147, 151
Code page 16, 27, 28, 306
GB2312 285
ISO 8859 279, 280, 281, 282, 283
ISO Latin-1 284
JISX208 285
KSC5601 285
Microsoft 1250 280
Microsoft 1251 282
Microsoft 1252 280
Microsoft 1253 282
Microsoft 1254 283
Microsoft 1255 283
Microsoft 1257 281
Microsoft 874 284
Code page scanner 67
Code page tables 279
Code point 28
Common character set 16, 20
Communication
   Homogeneous 156, 161
   Inhomogeneous 156, 161
Communication code page 164
   Determining 165
Communication IDoc 187
Communication language 184
Communication scenario
   Unicode/MDMP 203
Communication technologies 177
Communication type 170
Communication with RFC 161
Consistency check 76
Consolidation plans 248
Container alignment 136, 140, 197
   For Asian Double-Byte languages 196
Container problem 137
Container technology 136, 197
Conversion errors 144
Conversion of complex landscapes
   Criteria 247
Conversion process 156
Conversion projects 234
Conversion sequence 236
Conversion strategies 236
Conversion times 96
CORBA 49
Cost estimate 242
Country customizing 269
Country-specific information 300
Coverage Analyzer 147, 150
CPU 306
Cryptography 307
CU & UC 111, 113, 249
Currencies 15
Customer namespace 191
Customizing
   Language-dependent 266
Customizing includes 140

D

Data transfer 24
   Multilingual 168
Data type
   LANG 168
Data types
   Character-based 163
   Character-type 131, 132
   Non-character-type 131
   Numeric 132
Database code page 29
DBCS 306
DDIC attribute
   Text language 71
Deep data types 163
Default set 287
Delta problems 88
Development freeze 246
Device types 228
Dialects 217
Distribution monitor 99
DMEE 173
Double Byte Character set 306
Double-Byte 34, 39
Double-Byte code page 16, 279
Double-Byte languages 132, 136, 197
Downtime 249
Dynamic code-page switch system 41
Index

E
EBCDIC 29, 306
ecATT 306
ECMAScript 49
EDID4-SDATA 136
EDIFACT 306
Effort category 219
Encoding
   Hexadecimal 28
   Encoding form 306
   Encoding scheme 306
   Endianess 52, 157, 162
   End-user training 246
   Enterprise Services Architecture 53
   Enterprise SOA 53, 54
   Estimate of effort 236
   EUC 306
   Evaluation phase 234
   Expansion
      In new countries 275
      Export 90
      Export control table 76
      Extension category 140
      Extractor jobs 213
      Extractors 180

F
F1 converter 163
Failover 306
Fallback code page 70
File interfaces 142
File transfer 142
Flat data type 163, 167
Font 306
Fool the system 38, 279
Front end 218
Front-end code page 29
FTP 306

G
GB2312 35
German ASCII 306
Global character set 32, 162
Global master data 23
Globalization 13
Glyph 306
Golden MDMP rules 45
GUI 306
GUI_UPLOAD/DOWNLOAD 175

H
Handshake 165
Hangul 306
Hexadecimal encoding 28
Hint 84
Homogeneous communication 156, 161
HPUTF8 226
HTML 307

I
I18N 223
I18N mode 101
I18N option 226
IDoc 136, 307
IDoc adapter 160
IDoc base type 187
IDoc data transfer
   Multilingual 197
   IDoc structure 188, 192, 193, 196, 198
   Language-dependent 192, 197
IMIG 99
Import 91
Inbound processing 186, 188, 191, 198,
   199, 201
INDX-type tables 69
Information acquisition 235
Inhomogeneous communication 156,
   160, 161, 197
Input method 220
Input method editor 218
Interface catalog 237
Interface technology 177
Interfaces 307
Minimization 247
Internal language key 225
International address versions 272
Internationalization 13
Internet Transaction Server 53, 307
ISCII 307
ISO 307
ISO 8859 27, 28, 32, 35, 39, 40
ISO Latin-1 16
ISO/IEC 10646 19
Index

ISO-639   218
IT solution
  Global    275
iView      54

J

J2EE      307
Java      21, 49
Java stack 95, 160, 167
JDBC      307

K

Kanji     307
Katakana  307
Keyboard layout 220
KSC5601   36

L

Landscape
  Mixed      58
LANG
  Data type  168
Language CD 264
Language determination
  Methods   83
Language import 256, 264, 266
Language installation 256
Language key 43, 218
  Internal  225
Language transport 262
Language vector 225
Language-dependent
  IDoc structure 192, 197
Language-dependent customizing 266
Language-dependent object 256
Languages 15
Latin-1 character set 32
Latin-4    307
LDAP       49
Legacy mode 143
Length access 134
Length programming 134
Letters 28
LEXUTF8   226
Little Endian 51, 307
Load balancing 307
Locale     40
Localization 13
Logon language 165, 174
Long text editor 261
Long texts
  Multilingual 187
LSB        307

M

Maintenance landscape 244
Markup     307
Master data
  Global    23
Master IDoc 187
Master language 113
MBCS       307
MDMP       37, 38, 41, 111
MDMP partner system 170
MDMP rules
  Golden    45
MDMP system 41, 45
MDMP/MDMP communication 164
MDSP system 40
Message broker 308
Message queuing 308
Message type 188, 193, 195, 204
Microsoft MS 1250 29
Microsoft MS 1251 27
Middleware 156, 182, 308
Migration monitor 97
Mixed code page system 41
MNLS       39
MNLS system 38
Modification 149
Modification degree 185
MSB        308
Multi Byte character set 308
Multi Byte languages 34, 197
Multilingual ALE message types 202
Multilingual data transfer 168
Multilingual IDoc data transfer 197
Multilingual long texts 187

N

Nametab 65
New installation 233
NLS_GET_LANGLU_CP_TAB  139
NNTP  308
Non-Unicode system  129

O
Object
  Language-dependent  256
Object time  15
Octet  308
ODBC  308
Offset access  134
Offset programming  134
Online repository texts  261
OS  308
OS/DB migration  61, 244, 275
Outbound processing  186

P
Partner systems  158
Patch  308
Patterns  83
PC UI  54
PCL  308
Peripheral code pages  29
Pl components  204
Plain text  308
Plug-in components  204
Portal  308
PREPARE  115
Print drivers  221
Printed characters  28
Printing
  In the Unicode system  106
Production system  303
Programs
  Fenerated  149
  Generated  149
Project duration  241
Project runtime  242
Proof of concept  238
Proposal pool  260

Q
qRFC  180, 204

R
R3load  91, 126
R3up  117
RADCUCNT  66, 116
RAID  308
RAM  308
Reference language  269
Release change  248
Replacement characters  144, 172
Reprocessing  87
RFC  308
RFC adapter  160
RFC bit option  184
RFC client  165
RFC communication  161
RFC communication code page  187
RFC connection  169
RFC conversion behavior  162
RFC function  168
RFC Unicode/non-Unicode communication  143
Rich text  308
RMI  308
Round-trip capable  112, 215
RSA6  211
RSCP_CONVERT_FILE  175
RSCPINST  223
RSREFILL  268
Runtime optimization  239
RZ10  148, 225

S
Sandbox conversion  243
SAP .NET Connector  160
SAP applications
  Unicode capability  61
SAP Business Suite  111
SAP Country Versions  14
SAP ECC  6.0  110
SAP Employee Self-Services  112
SAP ERP  2004  111
SAP ERP  2005  110, 111
SAP Globalization Services  300
SAP GUI  100, 144, 175
  Release 6.40  175
SAP Java Connector  160
SAP language package  254
Index

SAP logon 101
SAP Manager Self-Services 112
SAP NetWeaver 2004s 110
SAP NetWeaver Exchange Infrastructure 160
SAP R/3 Enterprise 111
SAP Smart Forms 221
SAP Transport Management System 261
SAP Unicode workshop 234
SAP Workplace 53
SAP_CODEPAGE 165
sapiconv 175
SAPinst 126
saplpd 228
SAPscript 221
SAPup 117
SAPWIN 228
SBCS 308
SCI 151
SCOV 150
Script 308
SE63 222, 226, 260, 261
Secure Network Communication 46
Semiautomatic translation 260
Service-oriented architecture 309
Services 234
Session language 165
SET LOCALE 174
setlocale() 40
SGML 309
Shift-JIS 309
Short text editor 261
Single code page systems 17, 32
Single Sign-On 46
Single-Byte code page 16, 279
Sizing 309
SJIS 35, 40, 309
SJIS code page 39
SLO 309
SM30 226
SM59 143, 169
SMLT 226, 264, 267
SMLT_EX 263
SNC 46
SOA 309
Solution scenarios for Unicode/MDMP communication 185
SPAM 264
SPAU 115
SPDD 115
Special characters 16, 279
Special options 169
SPRAS 44
SPUM4 115, 249
SPUMG 68, 114
SQL 309
SSO 46
Standard ALE transfer 187
Statistics 260
Step language 213
Structures 132
Character-type 132
Deep 132
Flat 132
Nested 132
Non-nested 132
SUMG 68, 122
Supplementation 266
Supplementation language 256, 268, 269
Supplementation logic 267
Support package 63, 202
Surrogate area 50
SY-LANGU 174
Synchronous transmission 309
System code page 29, 35, 174
System consolidation 251
System landscape
  Simplification 251
System time 15
System vocabulary 69, 78

T

Table split 98
Table structures 167
TABLES 168
Target code page 163
TCP/IP 309
TCPOC 164, 165
TECHED_UNICODE_EXERCISE 150
TECHED_UNICODE_SOLUTION 150
Text language
  DDIC attribute 71
Text language flag 168
Third-party software 233
Time zones 15
Index

TIS620  36
TOOLIMP4_UCMIG  118
Transfer structure  180
Translation
  Manual  271
  Of business data  253
  Semi-automatic  260
Translation gaps  266
Translation strategy  220, 222
Translation tools  259
Translation transports  262
Translation Workbench  66, 261
Transmission
  Asynchronous  305
  Synchronous  309
Transport
  Unicode/non-Unicode  107
tRFC  186, 187
tRFC connection  197
TU & UC  111, 251
Twin system  251
Two-server methods  97

U

UCHECK  115, 147, 238
UCMIG_REQINC  118
UCMIG_STATUSCHK1  120
UCS  309
UM4_TXFLAG UPLOAD  120
Unambiguos blended code page  39
Unicode
  New installation  233
  SAP notes  301
Unicode architecture  57
Unicode capability
  SAP applications  61
Unicode code page  19
Unicode code point  28
Unicode Consortium  22, 300
Unicode conversion  60, 61, 304
  Documents  303
  Hardware requirements  63
  Reasons  241
  Typical process  303
Unicode enabling  238
Unicode encoding forms  20, 309
Unicode encoding schemes  20, 50, 59, 309
Unicode flag  129
Unicode font  227
Unicode nametabs  114
Unicode printers  107, 221, 226
Unicode project  300
Unicode scalar value  28
Unicode standard  19, 22
Unicode strategy  235
Unicode system  129
  Languages  287
Unicode Transformation Format  20, 52, 309
Unicode versions  22
Unicode/MDMP communication
  solution scenarios  185
Unicode/MDMP communicationscenario  203
Unicode-specific introductions  300
Upgrade  202
User exit  149, 206
User time  15
UTF  309
UTF-16  20, 51, 156
UTF-8  20, 51, 156
  XML intermediate format  163
Utilities classes  144
UUEncode  309

V

VAS  221
Vietnamese character set  221
Vocabulary maintenance
  Variants  86
Vocabulary transfer  82
Voice over IP  310

W

W3C  310
WE8DEC  29
WML  49, 310
Worklist  260
Wrapper function  193
WS DOWNLOAD  193
WS UPLOAD  144
WSDL  310
Index

X

XBRL 310
XHTML 310
XML 49, 310
XRFC 163

Z

Zcsa/installed_languages 225