

Making Programs Unicode Enabled

An Overview

TD Core AS&DM I18N Technology (AG)

May 15, 2009

THE BEST-RUN BUSINESSES RUN SAP™



Agenda



1. **ABAP Language Enhancements**
2. File Interfaces
3. Communication via RFC
4. Unicode Enabling Tools

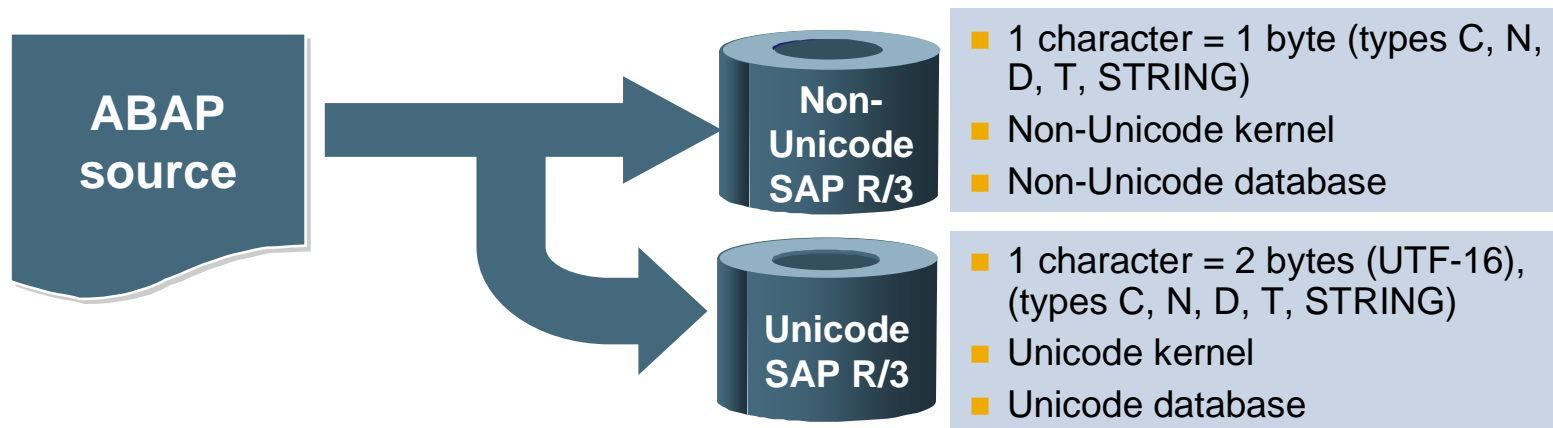
Planning: Unicode Enabling of Customer ABAP Programs



Unicode Enabling must be performed before the Unicode conversion in the Non-Unicode system via transaction UCCHECK (available as of SAP Web AS 6.20).

In UCCHECK a clear distinction between character and byte processing in Unicode is made.

Result: Same ABAP source in Unicode and in non-Unicode system



Representation of Unicode Characters



UTF-16 – Unicode Transformation Format, 16 bit encoding

- Fixed length, 1 character = 2 bytes (surrogate pairs = 2 + 2 bytes)
- Platform dependent byte order

UTF-8 – Unicode Transformation Format, 8 bit encoding

- Variable length, 1 character = 1...4 bytes
- Platform independent
- 7 bit US ASCII compatible

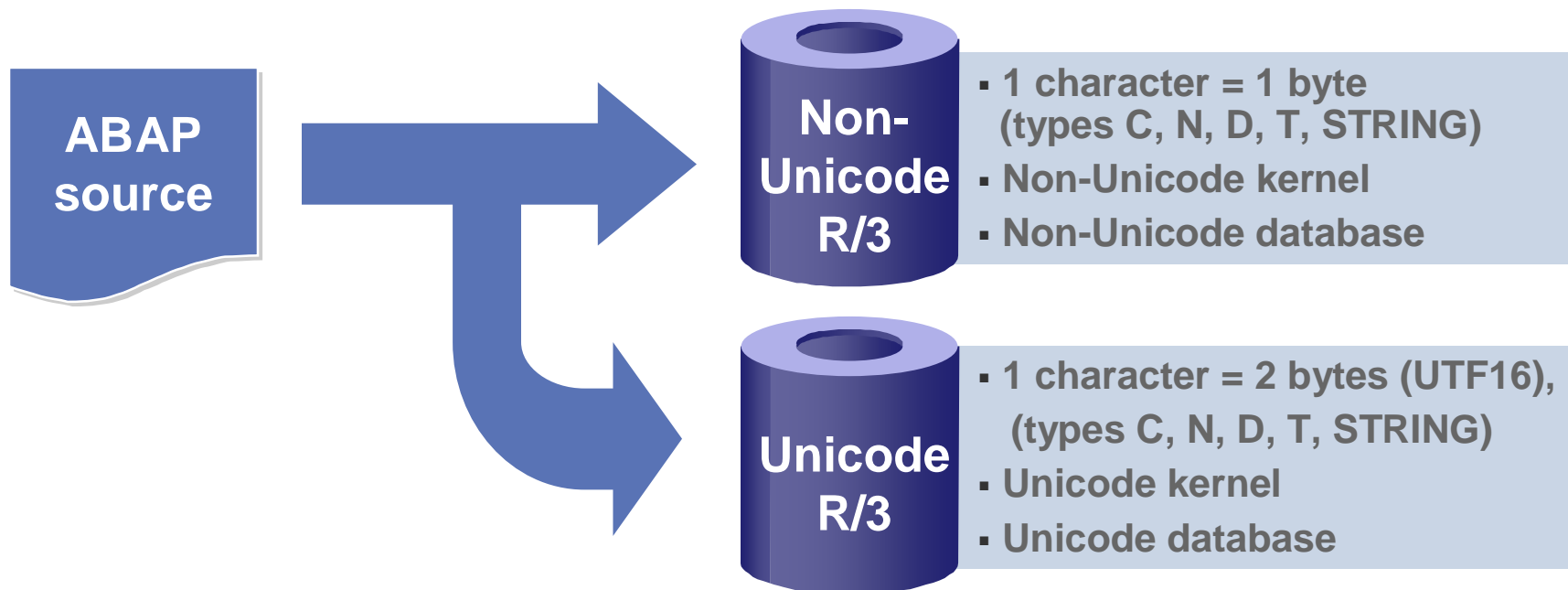
Character	Unicode scalar value	UTF-16 big endian	UTF-16 little endian	UTF-8
a	U+0061	00 61	61 00	61
ä	U+00E4	00 E4	E4 00	C3 A4
α	U+03B1	03 B1	B1 03	CE B1
會	U+3479	34 79	79 34	E3 91 B9
中石 中碯	U+2007B	DA00 DC7B	00DA7BDC	F0A081BB

- **Keep existing coding as far as possible**
- **Use existing character type for Unicode data**
- **Use UTF-16 to keep buffer sizes**



Character Expansion Model

- Separate Unicode and non-Unicode versions of R/3



- No explicit Unicode data type in ABAP
- Single ABAP source for Unicode and non-Unicode systems

Distinguish character and byte-based data

1 Character ≠ 1 Byte

Agenda



1. ABAP Language Enhancements
- 2. File Interfaces**
3. Communication via RFC
4. Unicode Enabling Tools

Program Attribute „Unicode checks active“



Title	Example: Unicode enabled program		
Original language	EN	English	
Created	13.08.2001	SCHIED	
Last changed by	13.08.2001	SCHIED	
Status	Inactive		

Attributes	
Type	Executable program
Status	
Application	
Authorization Group	
Development class	\$TMP Private Test Programs and Utilities
Logical database	
Selection screen	
<input type="checkbox"/> Editor lock	<input checked="" type="checkbox"/> Fixed point arithmetic
<input checked="" type="checkbox"/> Unicode checks active	<input type="checkbox"/> Start using variant

Save [Pencil icon] [Magnifying glass icon] [Share icon] [Close icon]



Program attribute „Unicode checks active“

- Required to run on a Unicode system

	Non-Unicode system	Unicode system
Attribute set (Unicode enabled)	ok	ok
Attribute not set (not Unicode enabled)	ok	not allowed

- If attribute is set, additional restrictions:
 - apply at compile and at run time
 - apply in Unicode systems and in non-Unicode systems
 - ensure that program will run on non-Unicode and Unicode systems with (almost) identical behavior

Character Processing

```
CONCATENATE cf1 cf2 TO cf3.  
IF cf1 CS cf2. ...
```

- String operations are only allowed for character-like operands
 - ABAP types C, N, D, and T, STRING
 - Structures consisting only of characters (C, N, D, T)
 - X and XSTRING are no longer considered character-like types

Byte Processing

```
CONCATENATE xf1 xf2 TO xf3 IN BYTE MODE.  
IF xf1 BYTE-CS xf2. ...
```

- Variants of string operations for byte processing
 - Addition „IN BYTE MODE“ for statements
 - Prefix „BYTE-“ for comparison operations
- Only operands of type X or XSTRING allowed

Determining the Length and Distance

- Counted in bytes or in characters? Specify!

DESCRIBE FIELD...LENGTH... IN (BYTE | CHARACTER) MODE

DESCRIBE DISTANCE BETWEEN ... AND ... INTO ...
IN (BYTE | CHARACTER) MODE.

Example

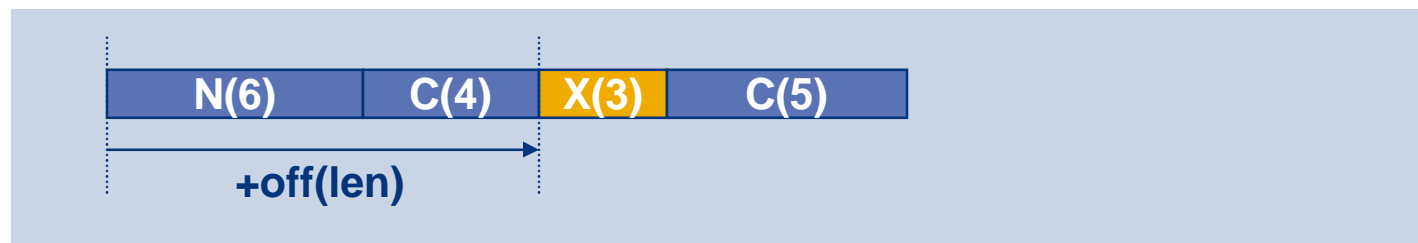
```
FORM write3 USING fld TYPE c.  
  DATA: fldlen TYPE i.  
  DESCRIBE FIELD fld LENGTH fldlen IN CHARACTER MODE.  
  IF fldlen >= 3.  
    WRITE: / fld(3).  
  ENDIF.  
ENDFORM.
```

Unicode Restrictions - Access with Offset or Length



Access To Structures With Offset/Length

- Structure must begin with characters
- Offset/length counted in characters
- Access only allowed within the character type prefix of a structure



ASSIGN fld+off(len) TO ...

- Access must not exceed field boundaries
- If ASSIGN fails, field-symbol is set to „unassigned“
- New ... RANGE addition allows the permissible boundaries to be expanded

New ABAP Features – Includes with Group Names

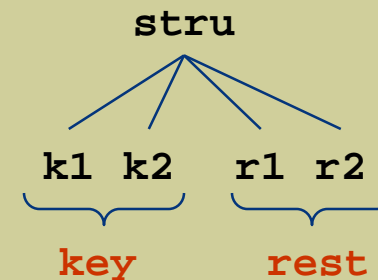


Symbolic Access to Includes of Structures

```
TYPES: BEGIN OF t_key,  
       k1(2) TYPE x,  
       k2(2) TYPE c,  
       END OF t_key.
```

```
TYPES: BEGIN OF t_rest,  
       r1(10) TYPE c,  
       r2(10) TYPE c,  
       END OF t_rest.
```

```
DATA: BEGIN OF stru.  
      INCLUDE TYPE t_key as key.  
      INCLUDE TYPE t_rest as rest.  
DATA: END OF stru.  
DATA: skey TYPE t_key,  
      srest TYPE t_rest.
```



Pre-Unicode

```
skey = stru(4).  
srest = stru+4(20).  
WRITE: stru-r2.
```

Unicode enabled with group names

```
skey = stru-key.  
srest = stru-rest.  
WRITE: stru-r2.
```

Think in types

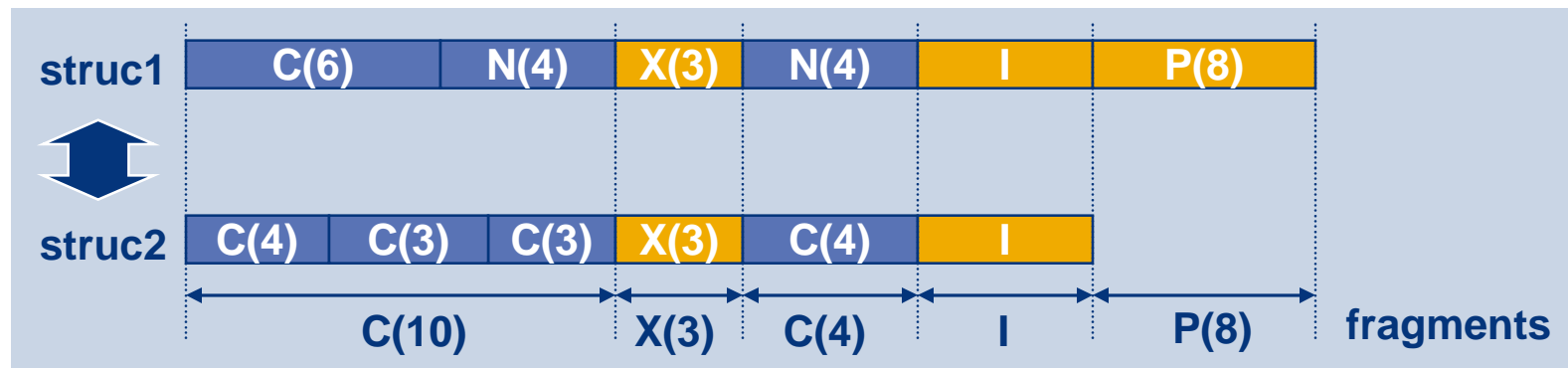
Think in semantics

Tell the system what you want to do



MOVE Between Incompatible Structures

- Matching data layout („fragment views“) required



Example

```
DATA:
  BEGIN OF cstru,
    first(10) TYPE c,
    tab(1)    TYPE c,
    last(10)  TYPE c,
  END OF cstru.
```

cstru = xstru.

```
DATA:
  BEGIN OF xstru,
    first(10) TYPE c,
    tab(1)    TYPE x VALUE '09',
    last(10)  TYPE c,
  END OF xstru.
```

"Unicode error!"



Using fields of type xstring as data containers

■ Writing data to an xstring

```
DATA: my_buffer TYPE xstring.  
      data1      TYPE some_type.  
  
...  
  
EXPORT id = data1 TO DATA BUFFER my_buffer.
```

- Data is stored in a platform-independent format
- Contents of xstring can be exchanged with any other 6.10-system (Unicode and non-Unicode)

■ Reading data from an xstring

```
FORM read_buffer USING buffer TYPE xstring.  
  DATA: fld2 TYPE some_type.  
  IMPORT id = fld2 FROM DATA BUFFER buffer.  
  
...  
ENDFORM.
```

- Automatic conversion of data during import



Creating data objects dynamically

- Creating and accessing data objects on the heap

```
DATA: dref TYPE REF TO data.  
CREATE DATA dref TYPE sometype.  
CREATE DATA dref TYPE (typename).  
CREATE DATA dref TYPE c LENGTH len.  
CREATE DATA dref TYPE STANDARD TABLE OF (typename)  
ASSIGN dref->* TO <f>.           "access data object
```

Casting to user-defined types

- Look at the contents of a field as a value of another type

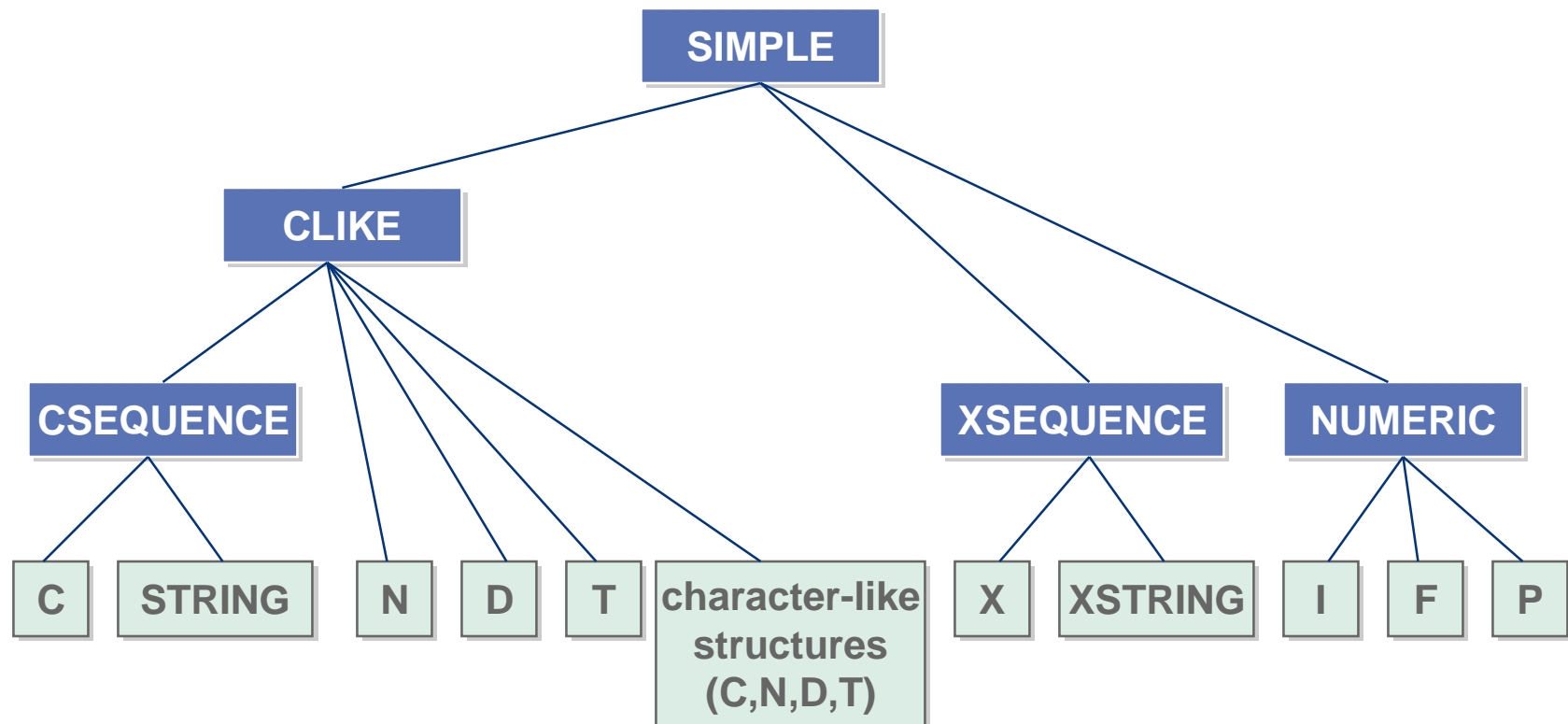
```
FIELD-SYMBOLS: <f> TYPE any.  
ASSIGN fld TO <f> CASTING TYPE sometype.  
ASSIGN fld TO <f> CASTING TYPE (typename).
```

- fld must provide sufficient alignment and length for the given type



New generic types for parameters and field-symbols

- Eliminate untyped parameters or field-symbols for improved security and performance



New ABAP Features – Enhancement categorization



If you are writing software for others you may have the following

Problem

- Enhancements on structures or tables may affect your coding:
 - Syntax-/runtime errors
 - Changed behavior (e.g. damaged or changed data)

Solution

- Maintaining the enhancement category in the DDIC: SE11 (Extras -> Enhancement Category)
 - Can not be enhanced
 - Can be enhanced - character like
 - Can be enhanced – character and numerical type
 - Can be arbitrarily enhanced
- Additional checks are done on your ABAP programs (SLIN) and show possible problems in allowed enhancement situations



ABAP lists: Difference between memory and display length

'한'	Character units in the memory	Display columns
Non-Unicode	2	2
Unicode	1	2

→ *1 Character ≠ 1 Display Column*

New ABAP Features - Different List Types



Half width (Default)

Demoprogram ABA 1

KO	하나	1
EN	one	1
KO	둘	2
EN	two	2
KO	셋	3
EN	three	3
KO	넷	4
EN	four	4
KO	다섯	5
EN	five	5
KO	여섯	6
EN	six	6
KO	일곱	7
EN	seven	7
KO	여덟	8
EN	eight	8
KO	아홉	9
EN	nine	9
KO	열	10
EN	ten	10
KO	스물>	21
KO	서른>	37

Dynamic

Demoprogram ABA 1

KO	하나	1	
EN	one	1	
KO	둘	2	
EN	two	2	
KO	셋	3	
EN	three	3	
KO	넷	4	
EN	four	4	
KO	다섯	5	
EN	five	5	
KO	여섯	6	
EN	six	6	
KO	일곱	7	
EN	seven	7	
KO	여덟	8	
EN	eight	8	
KO	아홉	9	
EN	nine	9	
KO	열	10	
EN	ten	10	
KO	스물	하나	21
KO	서른	일곱	37

Full width

Demoprogram

KO	하나	1	
EN	one	1	
KO	둘	2	
EN	two	2	
KO	셋	3	
EN	three	3	
KO	넷	4	
EN	four	4	
KO	다섯	5	
EN	five	5	
KO	여섯	6	
EN	six	6	
KO	일곱	7	
EN	seven	7	
KO	여덟	8	
EN	eight	8	
KO	아홉	9	
EN	nine	9	
KO	열	10	
EN	ten	10	
KO	스물	하나	21
KO	서른	일곱	37

Use ALV Grid and ALV List

Explicitely define display length

**Class CL_ABAP_LIST_UTILITIES
helps in complex cases**

Class CL_ABAP_CHAR_UTILITIES

- Constant attributes with system specific values

charsize length of 1 character in bytes

newline

cr_lf

form_feed

horizontal_tab

vertical_tab

backspace

minchar X'00' in non-Unicode systems, U+0000 in Unicode systems

maxchar X'FF' in non-Unicode systems, U+FFFD in Unicode systems

Example

```
CLASS cl_abap_char_utilities DEFINITION LOAD.  
DATA: text TYPE string.  
REPLACE cl_abap_char_utilites=>horizontal_tab  
        WITH space INTO text.
```




Conversion Classes

- Code page conversion
 - Unicode / non-Unicode code pages
- Endian conversion
 - little endian / big endian byte order
- Character conversion
 - Unicode codepoint / ABAP character

ABAP Class	Conversion
CL_ABAP_CONV_IN_CE	any code page → system code page
CL_ABAP_CONV_OUT_CE	system code page → any code page
CL_ABAP_CONV_X2X_CE	any code page → any code page

Agenda



1. ABAP Language Enhancements
2. File Interfaces
- 3. Communication via RFC**
4. Unicode Enabling Tools

**Know the code page used by the other side
or ask the user**

Pattern for writing/reading files on the application server:

OPEN DATASET IN <mode> MODE

TRANSFER/READ

CLOSE DATASET

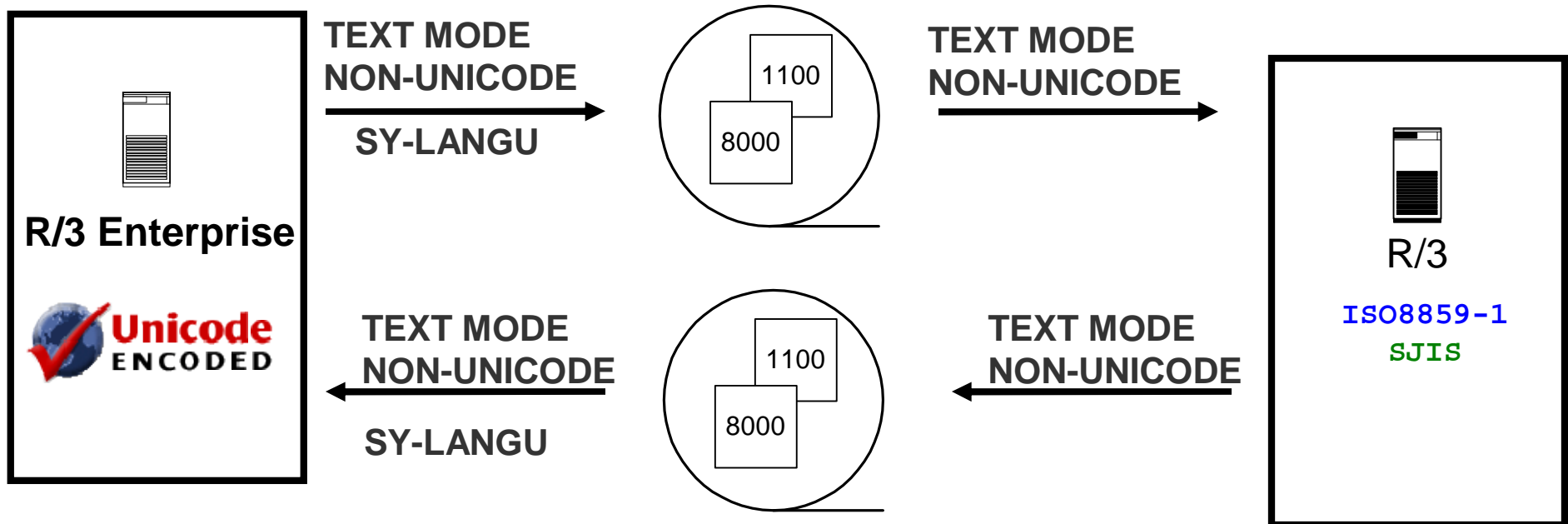
TEXT MODE ENCODING NON-UNICODE

- Allowed types
character like
- Behavior
convert text data between system code page and non-Unicode encoding matching to current system language (sy-langu)
- Usage
backward compatible exchange of text data with systems that cannot support UTF-8

File Transfer: TEXT MODE ENCODING NON-UNICODE



Example: TEXT MODE NON-UNICODE



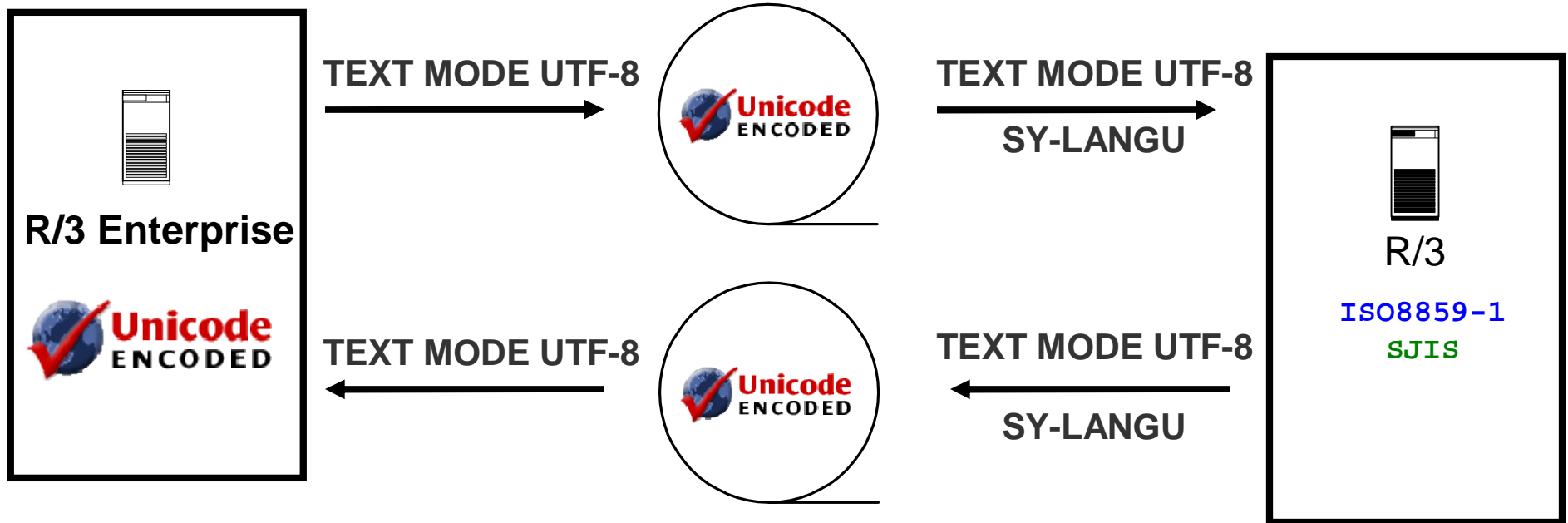
- ☹️ Only part of UC charset supported (possible data loss in the file)
- ☹️ Structured data as a whole → write field by field = 😊

TEXT MODE ENCODING UTF-8

- Allowed types
character like
- Behavior
convert text data between system code page and UTF-8 encoding
- Usage
loss free exchange of text data



Example: TEXT MODE UTF-8



☺ Full charset supported (no data loss in the file)

☹ Structured data as a whole → write field by field = ☺

BINARY MODE

- Allowed types
binary and character like
- Behavior
no conversion, copy bytes between memory and file
- Usage
fix format with defined encoding, endian and data structure

File Transfer: Using XML



Using XML as transport format

Use CALL TRANSFORMATION with target data type XSTRING to create an UTF-8 based XML representation of your data.

- Structure information (no layout / alignment problems)
- UTF-8 based (no data loss)
- Transport in binary form

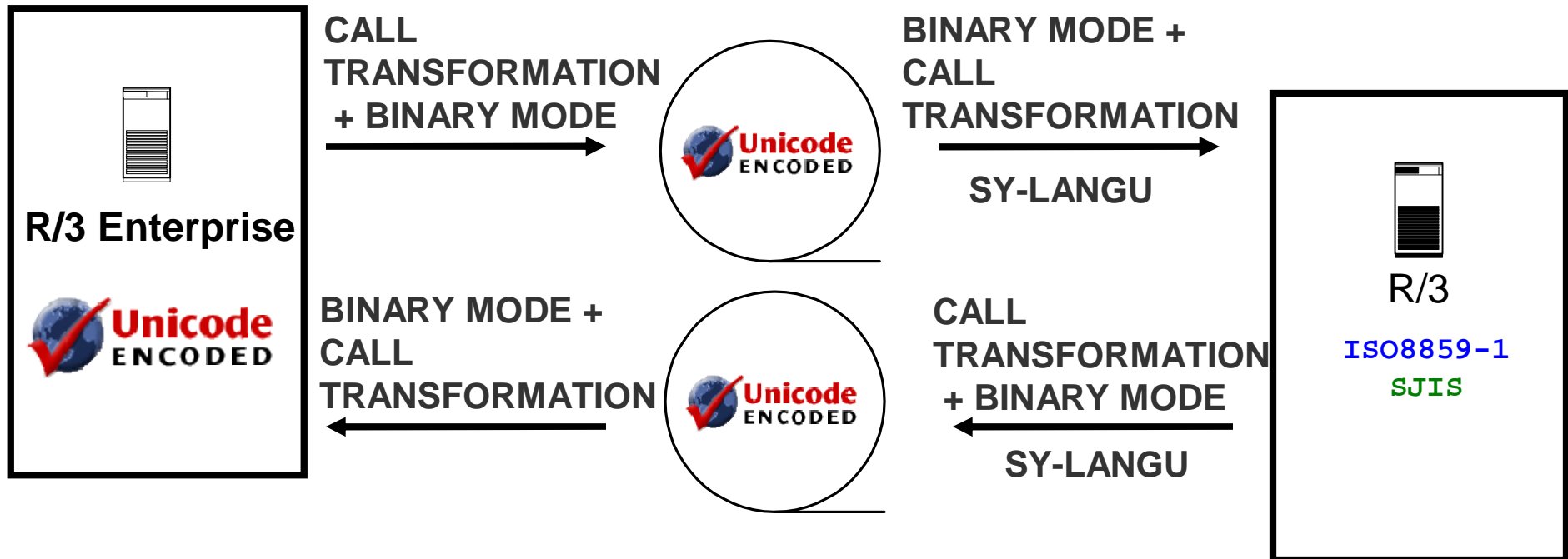
```
C:\perforce\bis\Unicode\presentation\xml\example.xml - Microsoft Internet Explorer provided by SAP IT
File Edit View Favorites Tools Help
Address Links

<?xml version="1.0" encoding="utf-8" ?>
- <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
- <asx:values>
- <COLORS>
- <TECHED03_COLORS>
  <COLOR>G</COLOR>
  <SPRAS>3</SPRAS>
  <NAME>초록색</NAME>
  <RGB>AP8A</RGB>
</TECHED03_COLORS>
- <TECHED03_COLORS>
  <COLOR>G</COLOR>
  <SPRAS>D</SPRAS>
  <NAME>grün</NAME>
  <RGB>AP8A</RGB>
</TECHED03_COLORS>
- <TECHED03_COLORS>
  <COLOR>G</COLOR>
  <SPRAS>J</SPRAS>
  <NAME>緑</NAME>
  <RGB>AP8A</RGB>
</TECHED03_COLORS>
</COLORS>
</asx:values>
</asx:abap>
```

File Transfer: BINARY + Using XML



Example: UTF-8 based XML + BINARY MODE



- ☺ Full charset supported (no data loss in the file)
- ☺ Structured data

LEGACY TEXT/BINARY MODE

- Allowed types
 - binary and character like
- Behavior
 - Produce in Unicode system format of non-Unicode
 - convert text data to non-Unicode
 - read/transfer binary data as is
- Usage
 - backward compatibility with systems that mix binary and character data within a file
- Difference between TEXT and BINARY
 - LEGACY TEXT MODE adds end of line marker

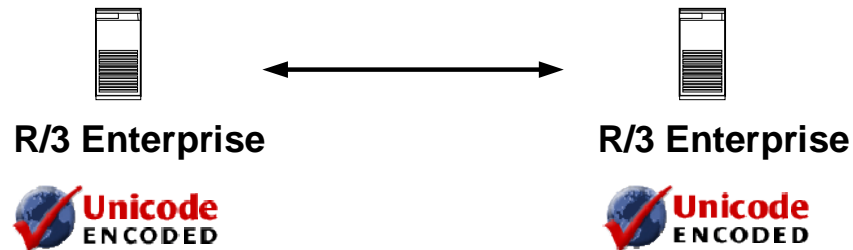
Agenda



1. ABAP Language Enhancements
2. File Interfaces
- 3. Communication via RFC**
4. Unicode Enabling Tools

Everything Unicode
or ensure data fit to receiver

RFC Unicode ↔ Unicode



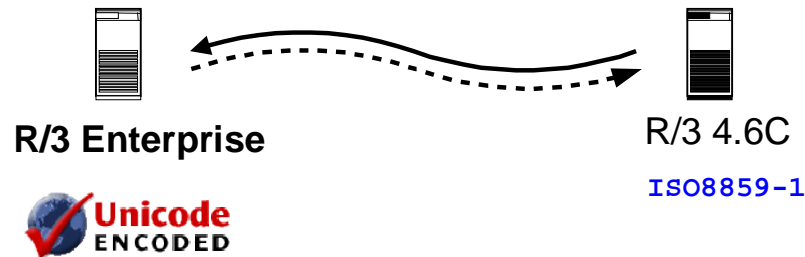
In case of an Unicode ↔ Unicode combination RFC passes all character data without code page conversion or merely with adaption of the endianness.

- UTF-16 big endian = SAP code page 4102
- UTF-16 little endian = SAP code page 4103

Information about the destination is maintained in SM59 →
→ special options → character width in target system

- 1 Byte = non-Unicode
- 2 Byte = Unicode

RFC Unicode ↔ non-Unicode single code page

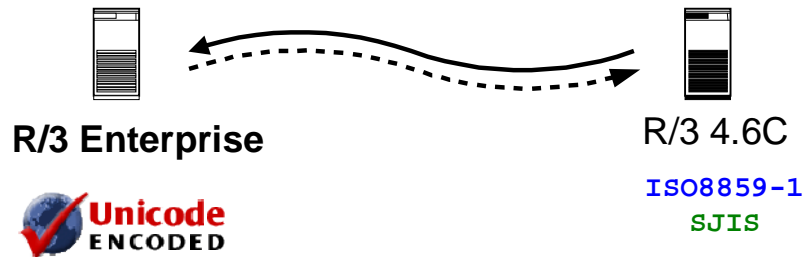


In case of an Unicode ↔ non-Unicode single code page combination, RFC passes all character data with code page conversion between Unicode and the old code page.

As Unicode is a true superset of any old standard codepage not all Unicode characters can be transferred to the non-Unicode system:

Ä	↔	Ä
ß	↔	ß
あ	→	#
東	→	#
한	→	#
₩	→	#

RFC Unicode ↔ Non-Unicode MDMP (Table with Language Key)



In case of an Unicode ↔ non-Unicode MDMP combination RFC passes all character data with code page conversion between Unicode and the different old code pages.

Which of the MDMP code pages is chosen depends on the language:

Ä	← DE →	Ä
ß	← DE →	ß
あ	← JA →	あ
東	← JA →	東

This mechanism is only available for flat tables with language key

Agenda



1. ABAP Language Enhancements
2. File Interfaces
3. Communication via RFC
4. **Unicode Enabling Tools**

Step 1

In **non-Unicode** system:

- Adapt all ABAP programs to Unicode syntax and runtime restrictions
- Set attribute "Unicode enabled" for all programs

Step 2

- Set up a Unicode system
 - Unicode kernel + Unicode database
 - Only ABAP programs with the Unicode attribute are executable
- Do **runtime tests in Unicode system**
 - Check for runtime errors
 - Look for semantic errors
 - Check ABAP list layout with former double byte characters

Transaction UCHECK



Program Edit Goto System Help

Check a Program Set for Syntax Errors in Unicode Environment

Docu for ABAP + Unicode UCCHECK Documentation

Object Selection

Object name	ZTECHED_UNICODE_E...	to		➔
Object Type		to		➔
Author (TADIR)		to		➔
Package		to		➔
Original system		to		➔

Check only programs where the Unicode flag is unchecked
 Include only Objects with Object Repository Entry (TADIR)
 Exclude \$* Packages

Restriction of Program Set to Prevent Timeout

Maximum Number of Programs 50

Statements that cannot be analysed statically

Display lines that cannot be analyzed statically
 Show also Locations Hidden with "#EC *
Includes to Be Displayed LSVIM* to ➔

Application-specific Checks

View Maintenance
 Obsolete Function Modules: UPLOAD/DOWNLOAD

Step 1 – Unicode Enabling with UCCHECK



Use UCCHECK to analyze your applications:

- Remove errors
- Inspect statically not analyzable places (optional)
 - Untyped field symbols
 - Offset with variable length
 - Generic access to database tables
- Set unicode program attribute using UCCHECK or SE38 / SE24 / ...
- Do additional checks with SLIN (e.g. matching of actual and formal parameters in function modules)

UCCHECK - Setting Unicode Flag



System Help

SAP

Result of Unicode Syntax Check

Ex...	Program	Include	Row	Message
■	ZTECHED_UNICODE_EXERCISE_4	ZTECHED_...	60	"ENTRY-CONTENTS" and "PERS" are not mutually convertible in a Unicode program.
■	ZTECHED_UNICODE_EXERCISE_4	ZTECHED_...	84	"ENTRY-CONTENTS" and "PERS" are not mutually convertible in a Unicode program.
■	ZTECHED_UNICODE_EXERCISE_4	ZTECHED_...	104	"PERS" and "CONTAINER_LINE-CONTENTS" are not mutually convertible in a Unicode
■	ZTECHED_UNICODE_EXERCISE_4	ZTECHED_...	104	program.
▲	ZTECHED_UNICODE_EXERCISE_5	ZTECHED_...	42	The system could not perform a static compatibility check on the current
▲	ZTECHED_UNICODE_EXERCISE_5	ZTECHED_...	42	statement, because of untyped or generic operands. The system will only carry
▲	ZTECHED_UNICODE_EXERCISE_5	ZTECHED_...	42	out this check at runtime.
▲	ZTECHED_UNICODE_EXERCISE_5	ZTECHED_...	45	The system cannot perform a static check on the validity of the offset/ length
▲	ZTECHED_UNICODE_EXERCISE_5	ZTECHED_...	45	entries for operand "CONTAINER+<DT>-OFFSET(<DT>-INTLEN)". They will be checked
▲	ZTECHED_UNICODE_EXERCISE_5	ZTECHED_...	45	at runtime.
▲	ZTECHED_UNICODE_EXERCISE_5	ZTECHED_...	48	The system cannot perform a static check on incompletely-typed operand "<F>" to
▲	ZTECHED_UNICODE_EXERCISE_5	ZTECHED_...	48	check whether or not the operand can be converted to a string. - -
●	ZTECHED_UNICODE_SOLUTION_1	ZTECHED_...	0	The system found no Unicode syntax errors
●	ZTECHED_UNICODE_SOLUTION_2	ZTECHED_...	0	The system found no Unicode syntax errors
●	ZTECHED_UNICODE_SOLUTION_3	ZTECHED_...	0	The system found no Unicode syntax errors
●	ZTECHED_UNICODE_SOLUTION_4	ZTECHED_...	0	The system found no Unicode syntax errors

What to do with the places that can only be checked at runtime?

- Reduce their number
 - In many cases you can specify the type of parameters and field-symbols
 - Use generic ABAP types where necessary
 - Mark those places that really need untyped parameters due to some kind of generic programming with “#EC * as OK after you did revise them.

- Do → Runtime tests

Final tests in the Unicode system

- Runtime tests, Runtime tests, Runtime tests, ...
 - Because the amount of warnings due to statically not analyzable places may be very large, you cannot type everything. In this case you have to rely on run-time tests.
 - Some semantic problems may be seen only in the Unicode system (e.g. byte or character length)
 - ABAP list layout can be checked only manually

Monitoring of runtime tests:

Having test plans is good, knowing the coverage of the test is better:

Use the ABAP Coverage Analyzer to monitor runtime tests

- **Distinguish characters and bytes**
- **Distinguish characters and display cells**
- **Think in types, think in semantics,
tell the system what you want**
- **Define code page for each communication**
- **You can start Unicode enabling today**



No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

SAP, R/3, xApps, xApp, SAP NetWeaver, Duet, SAP Business ByDesign, ByDesign, PartnerEdge and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned and associated logos displayed are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Einige von der SAP AG und deren Vertriebspartnern vertriebene Softwareprodukte können Softwarekomponenten umfassen, die Eigentum anderer Softwarehersteller sind.

SAP, R/3, xApps, xApp, SAP NetWeaver, Duet, SAP Business ByDesign, ByDesign, PartnerEdge und andere in diesem Dokument erwähnte SAP-Produkte und Services sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und in mehreren anderen Ländern weltweit. Alle anderen in diesem Dokument erwähnten Namen von Produkten und Services sowie die damit verbundenen Firmenlogos sind Marken der jeweiligen Unternehmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.

Die in diesem Dokument enthaltenen Informationen sind Eigentum von SAP. Dieses Dokument ist eine Vorabversion und unterliegt nicht Ihrer Lizenzvereinbarung oder einer anderen Vereinbarung mit SAP. Dieses Dokument enthält nur vorgesehene Strategien, Entwicklungen und Funktionen des SAP®-Produkts und ist für SAP nicht bindend, einen bestimmten Geschäftsweg, eine Produktstrategie bzw. -entwicklung einzuschlagen. SAP übernimmt keine Verantwortung für Fehler oder Auslassungen in diesen Materialien. SAP garantiert nicht die Richtigkeit oder Vollständigkeit der Informationen, Texte, Grafiken, Links oder anderer in diesen Materialien enthaltenen Elemente. Diese Publikation wird ohne jegliche Gewähr, weder ausdrücklich noch stillschweigend, bereitgestellt. Dies gilt u. a., aber nicht ausschließlich, hinsichtlich der Gewährleistung der Marktgängigkeit und der Eignung für einen bestimmten Zweck sowie für die Gewährleistung der Nichtverletzung geltenden Rechts.

SAP übernimmt keine Haftung für Schäden jeglicher Art, einschließlich und ohne Einschränkung für direkte, spezielle, indirekte oder Folgeschäden im Zusammenhang mit der Verwendung dieser Unterlagen. Diese Einschränkung gilt nicht bei Vorsatz oder grober Fahrlässigkeit.

Die gesetzliche Haftung bei Personenschäden oder die Produkthaftung bleibt unberührt. Die Informationen, auf die Sie möglicherweise über die in diesem Material enthaltenen Hotlinks zugreifen, unterliegen nicht dem Einfluss von SAP, und SAP unterstützt nicht die Nutzung von Internetseiten Dritter durch Sie und gibt keinerlei Gewährleistungen oder Zusagen über Internetseiten Dritter ab.

Alle Rechte vorbehalten.