

What is the Best Internationalization Approach to use?

It is assumed that you have read the chapter *Introduction to Internationalization* and are familiar with the various internationalization approaches described there.

This chapter provides information to help you decide which internationalization approach is the most appropriate. It covers the following topics:

- Conversion Overview
- Conversion Details
- Codepage Requirements for RPC Data Stream Conversions

See also *Configuring Broker for Internationalization* under z/OS | UNIX | Windows | BS2000/OSD | z/VSE.

Conversion Overview

This table gives an overview of the internationalization approaches that can be used. The approach you choose depends on

- ACI or RPC payload
- the type of codepage used by participants (client and server): single-byte or complex codepage configuration ⁽¹⁾, for example multibyte, double-byte, EBCDIC stateful codepages, Arabic shaping etc.

Internationalization Approach	Using Locale Strings	All components use single-byte codepages		One component uses a complex codepage configuration ⁽¹⁾		Usage Hint
		ACI	RPC ⁽²⁾	ACI	RPC ⁽²⁾	
<i>ICU Conversion</i>	yes ^{3,5}	yes	yes	yes	yes	<p>ICU conversion is recommended. In the Broker attribute file, set the service-specific or topic-specific broker attribute CONVERSION:</p> <ul style="list-style-type: none"> ● for <i>ACI-based Programming</i> to CONVERSION=SAGTCHA ● for <i>RPC-based Components and Reliable RPC</i> to CONVERSION=SAGTRPC <p>We recommend always using SAGTRPC for RPC data streams. <i>Conversion with Multibyte, Double-byte and other Complex Codepages</i> will always be correct, and <i>Conversion with Single-byte Codepages</i> is also efficient because SAGTRPC detects single-byte codepages automatically. See <i>Conversion Details</i>.</p> <p>See also <i>Configuring ICU Conversion</i> under z/OS UNIX Windows BS2000/OSD z/VSE under <i>Configuring Broker for Internationalization</i> in the platform-specific administration documentation.</p>

Internationalization Approach	Using Locale Strings	All components use single-byte codepages		One component uses a complex codepage configuration ⁽¹⁾		Usage Hint
		ACI	RPC ⁽²⁾	ACI	RPC ⁽²⁾	
<i>Translation</i>	no	yes	yes	no	no	<p>Translation is not recommended for the following reasons:</p> <ul style="list-style-type: none"> ● limited support of code points for ASCII, IBM EBCDIC and Fujitsu EBCDIC ● code points are not 100% compatible with standardized ASCII or EBCDIC codepages, which means that some code points are not roundtrip-compatible <p>Consider instead using ICU conversion, see first row in this table.</p>
<i>Translation User Exit</i>	no	yes	yes	yes	no	<p>Translation User Exit is not recommended. If you only wish to adapt code points, it is too much effort. We recommend you use ICU conversion instead. See <i>Translation User Exit Replacement with ICU Conversion</i>.</p>
<i>SAGTRPC User Exit</i>	optional ^{4,5}	no	yes	no	yes	<p>Requires considerable effort for implementation. See <i>Conversion Details</i>. Consider instead using ICU conversion. See first row in this table. Not available under z/VSE.</p>

Notes:

1. A complex codepage configuration is in effect where one participant (client or server) of a communication uses a codepage listed in *Conversion with Multibyte, Double-byte and other Complex Codepages*.
2. All codepages used for *RPC-based Components* and *Reliable RPC* must meet the *Codepage Requirements for RPC Data Stream Conversions*.
3. The locale string (codepage)

- must follow the rules described under *Locale String Mapping*
 - must be a codepage supported by the broker
 - must be the codepage used in your environment, otherwise unpredictable results may occur.
4. It depends on the implementation of the *SAGTRPC User Exit* whether locale strings (codepages) are used. See *Character Set and Codepage* under *z/OS | UNIX | Windows* in section *Configuring Broker for Internationalization*. If they are used, they must follow the rules described under *Locale String Mapping*.
 5. If the participant (client or server) does not send a codepage (locale string) you can optionally
 - set the *Codepage-specific Attributes* under *Broker Attributes* to meet your requirements, or
 - configure the participant (client or server). See *Preparing EntireX Components for Internationalization*.

Conversion Details

- Conversion with Single-byte Codepages
- Conversion with Multibyte, Double-byte and other Complex Codepages

Conversion with Single-byte Codepages

This table gives an overview of the conversion effort if two participants (client and server) of a communication use single-byte codepages only. It is valid for ICU conversion. For RPC, SAGTRPC detects single-byte codepages automatically and converts them efficiently in one step (a single ICU call) from source to target encoding. This is the same as SAGTCHA for ACI. The same applies if you have invented your own internationalization approach with *Translation User Exit*.

The effort does not depend on ACI or RPC payload - there is no difference. If one participant (client or server) uses a complex codepage configuration, the information given here does not apply; see *Conversion with Multibyte, Double-byte and other Complex Codepages* instead.

To find out if a codepage is single-byte, see *ICU Resources*.

Codepage Configuration	ACI ⁽¹⁾	RPC ⁽²⁾⁽³⁾
Single-byte codepages	Conversion is fast and efficient in one step.	Conversion is fast and efficient in one step.

Notes:

1. *ACI-based Programming*: in the Broker attribute file, the service-specific or topic-specific broker attribute CONVERSION is set to CONVERSION=SAGTCHA.
2. *RPC-based Components and Reliable RPC*: in the Broker attribute file, the service-specific broker attribute CONVERSION is set to CONVERSION=SAGTRPC.
3. All codepages used for *RPC-based Components and Reliable RPC* must meet the *Codepage Requirements for RPC Data Stream Conversions*.

Conversion with Multibyte, Double-byte and other Complex Codepages

This table gives an overview on the conversion effort if one participant (client or server) of a communication use a multi-byte, doublebyte or other complex codepage configuration (see the table), including Arabic shaping. It applies to ICU conversion. For RPC, SAGTRPC detects complex codepage configurations automatically and converts them as described (see column RPC) from source to target encoding. If you have invented your own internationalization approach with

- *Translation User Exit* for ACI, consider the rules in column ACI
- *SAGTRPC User Exit* for RPC, consider the rules in column RPC

depending on codepage type.

If two participants (client and server) of a communication use single-byte codepages only, see *Conversion with Single-byte Codepages*. With a complex codepage configuration, the effort depends on:

- ACI or RPC payload
- the type of codepage used: multi-byte, doublebyte or EBCDIC stateful, etc.
- whether Arabic shaping is required

To find out if a codepage is multibyte, double-byte or EBCDIC stateful, see *ICU Resources*.

Codepage Configuration	ACI ⁽¹⁾	RPC ⁽²⁾⁽³⁾
Multibyte or double-byte codepages	There is no additional effort compared to <i>Conversion with Single-byte Codepages</i> . Conversion is performed in one step, the same as with single-byte codepages. Please note the payload may change its length in bytes during conversion.	<p>If at least one participant (client or server) uses a multibyte or double-byte codepage with RPC, each IDL parameter (see <i>simple-parameter-definition</i>) must be converted separately. The data in IDL type A, AV, K and KV and RPC metadata may increase or decrease after conversion from the sender's source codepage to the receiver's target codepage. The following must be honored:</p> <ul style="list-style-type: none"> ● <i>increasing</i> or <i>decreasing</i> data within IDL type AV and KV (without maximum) and RPC metadata (such as user ID, IDL library and IDL program). ● <i>increasing</i> or <i>decreasing</i> data within IDL type A and K and AV, KV (with maximum) in its IDL defined field length boundaries. Data must be truncated if the field boundaries are crossed for increase - otherwise the RPC data stream is destroyed and unpredictable errors occur. If the data decreases, fields are padded at the end with blanks. <p>All other IDL data types are converted as with single-byte code pages.</p>

Codepage Configuration	ACI ⁽¹⁾	RPC ⁽²⁾⁽³⁾
EBCDIC stateful codepages, encoded with escape technique (SI/SO bytes)	There is no additional effort compared to <i>Conversion with Single-byte Codepages</i> . Conversion is performed in one step, the same as with single-byte codepages. Please note the payload may change its length in bytes during conversion. There is no special handling for SI/SO bytes as with RPC.	<p>If at least one participant (client or server) uses an EBCDIC stateful codepage with RPC, each IDL parameter (see <i>simple-parameter-definition</i>) must be converted separately. Also, the IDL types K and KV allow you to transfer double-byte data without SO and SI escape characters. This feature is designed for use in Asian countries. The disadvantage is that IDL fields must be converted field-by-field. To convert the fields correctly, RPC programmers have to consider the following rules, otherwise unpredictable results may occur:</p> <ul style="list-style-type: none"> ● SO and SI escape characters may not be contained in IDL type K and KV ● double-byte characters are allowed in IDL type K and KV only ● single-byte characters cannot be transferred in IDL type K and KV <p>All other IDL data types are converted as with single-byte code pages.</p>
Hebrew CP803 ⁽⁴⁾	There is no additional effort compared to <i>Conversion with Single-byte Codepages</i> . Conversion is performed in one step, the same as with single-byte codepages. Latin lowercase characters cannot be used and lead to conversion errors. See <i>OPTION Values for Conversion</i> to tune error behavior to meet your requirements.	<p>If at least one participant (client or server) uses the Hebrew codepage CP803, each IDL parameter (see <i>simple-parameter-definition</i>) must be converted separately, because CP803 does not include Latin lowercase characters ⁽³⁾. Please note the following:</p> <ul style="list-style-type: none"> ● All IDL types can be used. ● Latin lowercase characters cannot be used within IDL type A and AV. ● IDL program and IDL library cannot contain Latin lowercase characters, but Hebrew characters are OK. ● RPC error text, PING replies etc. are converted to uppercase before conversion to CP803. This makes such texts readable at both ends (clients and server).

Codepage Configuration	ACI ⁽¹⁾	RPC ⁽²⁾⁽³⁾
Arabic shaping ⁽⁵⁾	The additional effort compared to <i>Conversion with Single-byte Codepages</i> . The conversion itself is performed in one step, the same as with single-byte codepages. Shaping is performed on the complete ACI payload.	If Arabic shaping is required, each IDL parameter (see <i>simple-parameter-definition</i>) must be converted separately. Shaping is performed on IDL data types A, AV, K and KV. All other IDL data types are converted as with single-byte code pages.

Notes:

1. *ACI-based Programming*: in the Broker attribute file, the service-specific or topic-specific broker attribute CONVERSION is set to CONVERSION=SAGTCHA.
2. *RPC-based Components and Reliable RPC*: in the Broker attribute file, the service-specific broker attribute CONVERSION is set to CONVERSION=SAGTRPC.
3. All codepages used for *RPC-based Components and Reliable RPC* must meet the *Codepage Requirements for RPC Data Stream Conversions*.
4. The Hebrew CP 803 does not contain Latin lowercase characters and does not meet the *Codepage Requirements for RPC Data Stream Conversions*. Despite this non-compliance, it can still be used for RPC.
5. Arabic shaping is in effect if all participants (client and server) use one of the following codepages: UTF-8, windows-1256 or ibm-420 codepage.

Codepage Requirements for RPC Data Stream Conversions

Codepages used to convert RPC data streams must meet several requirements:

1. Codepages used to convert RPC data streams must have the following code points (characters) defined:

Character	also known as	Rendered	Unicode Code Point
uppercase letters A-Z without special characters		A - Z	0x0041 to 0x005A
lowercase letters a-z without special characters		a - z	0x0061 to 0x007A
digits		0-9	0x0030 to 0x0039
SPACE		" "	0x0020
LEFT PARENTHESIS	OPENING PARENTHESIS	"("	0x0028
RIGHT PARENTHESIS	CLOSING PARENTHESIS	")"	0x0029
PLUS SIGN		"+"	0x002B
HYPHEN	MINUS	"-"	0x002D
SOLIDUS	SLASH	"/"	0x002F
COLON		":"	0x003A
COMMA		","	0x002C
FULL STOP	PERIOD	"."	0x002E
EQUALS SIGN		"="	0x003D

2. All code points (characters) listed in the table above must have a unique mapping (without any fallbacks and reverse fallbacks) to/from Unicode, that is, they must be roundtrip-compatible.
3. If the codepage used is a multibyte or double-byte codepage, the code points (characters) listed in the table above must have a length of 1 byte within the codepage. Therefore UTF-16 encoding cannot be used, but UTF-8 encoding is possible.

Codepages that do not obey the rules above cannot be used for RPC-based components, because those code points (characters) are used to code for example the IDL library and IDL program, descriptive metadata and IDL type fields in numeric, integer and binary form.