

# Configuring Broker for Internationalization

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

This chapter explains in detail how to configure the broker for the various internationalization approaches, how to write a translation user exit and how to write a SAGTRPC user exit. It covers the following topics:

- Configuring Translation
- Configuring Translation User Exits
- Configuring ICU Conversion
- Writing Translation User Exits

See also *What is the Best Internationalization Approach to use?*

---

## Configuring Translation

### > To configure translation

- In the Broker attribute file, set the service-specific or topic-specific broker attribute TRANSLATION to SAGTCHA as the name of the translation routine. Example:

```
TRANSLATION=SAGTCHA
```

## Configuring Translation User Exits

### > To configure translation user exits

As a prerequisite, the user-written translation module must be accessible to the Broker worker threads.

1. Copy the user-written translation module into any sublibrary defined to EntireX Broker's LIBDEF chain.
2. In the Broker attribute file, set the service-specific or topic-specific broker attribute TRANSLATION to the name of the user-written translation routine. Example:

```
TRANSLATION=MYTRANS
```

## Configuring ICU Conversion

### > To configure ICU conversion

1. In the Broker attribute file, set the ICU-CONVERSION (default for z/VSE is NO):

```
ICU-CONVERSION=YES
```

2. In the Broker attribute file, set the service-specific attribute CONVERSION. Default for z/VSE is NO. Examples:

- ICU Conversion with SAGTCHA for *ACI-based Programming*:

```
CONVERSION=(SAGTCHA,TRACE=1,OPTION=SUBSTITUTE)
```

- ICU Conversion with SAGTRPC for *RPC-based Components and Reliable RPC*:

```
CONVERSION=(SAGTRPC,TRACE=2,OPTION=STOP)
```

We recommend always using SAGTRPC for RPC data streams. *Conversion with Multibyte, Double-byte and other Complex Codepages* will always be correct, and *Conversion with Single-byte Codepages* is also efficient because SAGTRPC detects single-byte codepages automatically. See *Conversion Details*.

3. Optionally configure a CONVERSION OPTION to tune error behavior to meet your requirements; see *OPTION Values for Conversion*.

### > To configure locale string defaults (optional)

- If the broker's locale string defaults do not match your requirements (see *Broker's Locale String Defaults*), we recommend you assign suitable locale string defaults for your country and region, see the respective attribute in *Codepage-specific Attributes under Broker Attributes* for how to customize the broker's locale string defaults.

### > To customize mapping of locale strings (optional)

- If the built-in locale string mapping mechanism does not match your requirements, you can assign specific codepages to locale strings. See *Broker's Built-in Locale String Mapping* and `locale-string` for information on customizing the mapping of locale strings to codepages.

## Writing Translation User Exits

This section covers the following topics:

- Introduction
- Structure of the TRAP Control Block
- Using the TRAP Fields

### Introduction

EntireX Broker provides an interface to enable user-written translation routines in the programming language . It contains three parameters:

- The address of the TRAP control block (TRAP = Translation Routine / Area for Parameters).
- The address of a temporary work area. It is aligned to fullword / long integer boundary (divisible by 4). The work area can only be used for temporary needs and is cleared after return.
- A fullword (long integer) that contains the length of the work area.

#### Note:

Names for user-written translation routines starting with "SAG" are reserved for Software AG usage and must not be used, e.g. "SAGTCHA" and "SAGTRPC".

### Structure of the TRAP Control Block

The Assembler dummy section TR\$TRAP covers the layout of the TRAP control block:

```

TR$TRAP  DSECT  ,
TR$TYPE  DS      F          TRAP type
TR$TYP2  EQU     2          TRAP type ETB 121
TR$ILEN  DS      F          Input buffer length
TR$IBUF  DS      A          Address of input buffer
TR$OLEN  DS      F          Output buffer length
TR$OBUF  DS      A          Address of output buffer
TR$DLEN  DS      F          Length of data returned:
*                               Should be set to the minimum value of TR$ILEN
*                               and TR$OLEN.
TR$SHOST DS      F          Sender's host:
*                               x'00000000' = little endian
*                               x'00000001' = big endian
TR$SCODE DS      F          Sender's character set:
SEBCIBM  EQU     X'00000022' EBCDIC (IBM)
SEBCSNI  EQU     X'00000042' EBCDIC (SNI)
SA88591  EQU     X'00000080' ASCII
TR$RHOST DS      F          Receiver's host      --> see TR$SHOST
TR$RCODE DS      F          Receiver's char set --> see TR$SCODE
TR$BHOST DS      F          BROKER host         --> see TR$SHOST
TR$BCODE DS      F          BROKER char set    --> see TR$SCODE
TR$SENV  DS      F          Sender's ENVIRONMENT field supplied:
OFF      EQU     X'00000000' ENVIRONMENT field not set
ON       EQU     X'00000001' ENVIRONMENT field set
*
TR$RENV  DS      F          Receiver's ENVIRONMENT field supplied:

```

```

*
TR$SENV DS CL32 --> see TR$SENV
TR$RENV DS CL32 Sender's ENVIRONMENT field
TR$LEN EQU *-TR$TRAP Receiver's ENVIRONMENT field
Length of TRAP

```

The translation routine USRTCHA is an example of the translation user exit, it is contained in the EntireX Common source library.

## Using the TRAP Fields

The TR\$DLEN must be supplied by the user-written translation routine. It tells the Broker the length of the message of the translation. In our example its value is set to the minimum length of the input and output buffer.

All other TRAP fields are supplied by the Broker and must not be modified by the user-written translation routine.

The incoming message is located in a buffer pointed to by TR\$IBUF. The length (not to be exceeded) is supplied in TR\$ILEN. The character set information from the send buffer can be taken from TR\$SCODE.

The outgoing message must be written to the buffer pointed to by TR\$OBUF. The length of the output buffer is given in the field TR\$OLEN. The character set is specified in TR\$RCODE. If the addresses given in TR\$IBUF and TR\$OBUF point to the same location, it is not necessary to copy the data from the input buffer to the output buffer.

The environment fields TR\$SENV and TR\$RENV are provided to handle site-dependent character set information. For the SEND and/or RECEIVE functions, you can specify data in the ENVIRONMENT field of the Broker ACI control block. This data is translated into the codepage of the platform where EntireX Broker is running (see field TR\$BCODE) and is available to the TR\$SENV or TR\$RENV field in the TRAP control block. TR\$SENV or TR\$RENV are set to ON if environmental data is available.

The sample source USRTCHA contains a section to handle the ENVIRONMENT value \*NONE. The translation will be skipped if \*NONE is supplied by the sender or receiver. Any values given in the API field ENVIRONMENT must correspond to the values handled in the translation routine.