

# Writing ACI Servers for the RPC-ACI Bridge in Natural

The RPC-ACI Bridge is prepared for ACI servers written in Natural.

This chapter covers the following topics:

- Tasks
  - Data Types
  - Declaring the Variables for the Data Types
- 

## Tasks

Writing an ACI server consists of two tasks:

- implement the Broker calls
- implement the processing of the received buffer and the response for the send buffer

## Using Arrays of Groups

If your programs use arrays of groups, you have to adjust the marshalling.

### ➤ To adjust the marshalling for arrays of groups

1. Use the property `entirex.rpcacibridge.marshalling` for the configuration.
2. Set the property to "natural".

If your programs do not use arrays of groups, you do not need to set `entirex.rpcacibridge.marshalling`.

## Data Types

Data Type	Description	Format	Note
<i>A</i> <i>number</i>	Alphanumeric	<i>number</i> bytes, encoding the characters.	
AV	Alphanumeric variable length	Bytes up to the end of the buffer.	1
AV[ <i>number</i> ]	Alphanumeric variable length with maximum length	Bytes up to the end of the buffer, maximal length <i>number</i> .	1
<i>K</i> <i>number</i>	Kanji	Same as data type A.	
KV	Kanji variable length	Same as data type AV.	1
KV[ <i>number</i> ]	Kanji variable length with maximum length	Same as data type AV[ <i>number</i> ].	1
I1	Integer (small)	<i>sign</i> (+, -) and 3 bytes (digits).	
I2	Integer (medium)	<i>sign</i> (+, -) and 5 bytes (digits).	
I4	Integer (large)	<i>sign</i> (+, -) and 10 bytes (digits).	
<i>N</i> <i>number1</i> [ . <i>number2</i> ]	Unpacked decimal	<i>sign</i> (+, -), <i>number1</i> bytes (digits) [ <i>number2</i> ] bytes (digits), no decimal point.	
<i>P</i> <i>number1</i> [ . <i>number2</i> ]	Packed decimal	<i>sign</i> (+, -), <i>number1</i> bytes (digits) [ <i>number2</i> ] bytes (digits), no decimal point.	
L	Logical	1 byte: X for true, all other false.	
D	Date	YYYYMMDD.	2
T	Time	YYYYMMDDhhmmssS.	3

### Notes:

1. Only as last value.
2. YYYY year, MM month, DD day.
3. YYYY year, MM month, DD day, hh hour, mm minute, ss second, S tenth of a second.

Data Types not supported:

- Binary (B[*n*], BV, BV[*n*])
- Floating point (F4, F8)

## Declaring the Variables for the Data Types

This section describes how to declare the variables for the data types. Use these declarations to map the receive buffer and the send buffer to variables. For some data types, the values have to be moved to a local variable before computation.

Example:

```
* Declaration
DEFINE DATA LOCAL
1 PNUMERIC (A012)
1 #NUMERIC (N8.3)
1 REDEFINE #NUMERIC
2 #NUMERIC1 (N11)
* Computation
    MOVE EDITED RCVE-DATA.PNUMERIC TO #NUMERIC1 (EM=S9(11))
    #NUMERIC := #NUMERIC + 1
    MOVE EDITED #NUMERIC1 (EM=S9(11)) TO SEND-DATA.PNUMERIC
```

Data Type	Declaration and Marshalling
A<number> Alphanumeric	Declaration for receive and send buffer: (An)
AV Alphanumeric variable length	Declaration for receive and send buffer: (A) DYNAMIC
AV[ <i>number</i> ] Alphanumeric variable length with maximum length	Declaration for receive and send buffer: (A) DYNAMIC
K<number> Kanji	Declaration for receive and send buffer: (An)
KV Kanji variable length	Declaration for receive and send buffer: (A) DYNAMIC
KV[ <i>number</i> ] Kanji variable length with maximum length	Declaration for receive and send buffer: (A) DYNAMIC
I1 Integer (small)	Declaration for receive and send buffer: (A4)MOVE EDITED to I1 variable with (EM=S9(3))
I2 Integer (medium)	Declaration for receive and send buffer: (A6)MOVE EDITED to I2 variable with (EM=S9(5))
I4 Integer (large)	Declaration for receive and send buffer: (A11)MOVE EDITED to I4 variable with (EM=S9(10))
N< <i>number1</i> >[. <i>number2</i> ] Unpacked decimal	Declaration for receive and send buffer: (An), where $n = \text{number1} + \text{number2} + 1$ (one byte for the sign). Redefine N <i>number1</i> + <i>number2</i> variable as N <i>number1</i> . <i>number2</i> variable. MOVE EDITED to N <i>number1</i> + <i>number2</i> variable with (EM=S9( <i>number1</i> + <i>number2</i> ))
P< <i>number1</i> >[. <i>number2</i> ] Packed decimal	Declaration for receive and send buffer: (An), where $n = \text{number1} + \text{number2} + 1$ (one byte for the sign). Redefine P <i>number1</i> + <i>number2</i> variable as P <i>number1</i> . <i>number2</i> variable. MOVE EDITED to P <i>number1</i> + <i>number2</i> variable with (EM=S9( <i>number1</i> + <i>number2</i> ))
L Logical	Declaration for receive and send buffer: (A1)
D Date	Declaration for receive and send buffer: (A8)MOVE EDITED to Date variable with (EM=YYYYMMDD)
T Time	Declaration for receive and send buffer: (A15)MOVE EDITED to Time variable with (EM=YYYYMMDDHHIISST)