

DBMS_MGWMSG

DBMS_MGWMSG provides object types used by the canonical message types to convert message bodies, and methods, constants, and subprograms for working with Messaging Gateway message types.

This chapter contains the following topics:

- [Security Model](#)
- [Constants](#)
- [Types](#)
- [Summary of DBMS_MGWMSG Subprograms](#)



See Also:

[DBMS_MGWADM](#) which describes the Messaging Gateway administrative interface, [DBMS_MGWADM](#)

DBMS_MGWMSG Security Model

The EXECUTE privilege is granted to PUBLIC on all types defined in the DBMS_MGWMSG package as well as the canonical types. The DBMS_MGWMSG packages and object types are owned by SYS.



Note:

You must run the `catmgw.sql` script to load the Messaging Gateway packages and object types into the database. Refer to the *Oracle Database Advanced Queuing User's Guide* for information on loading database objects and using DBMS_MGWMSG.

DBMS_MGWMSG Constants

The DBMS_MGWMSG package defines constants for specifying various parameter values.

Table 132-1 DBMS_MGWMSG Constants: Value Types and Constants Representing the Type of Value for a SYS.MGW_NAME_VALUE_T Object

Value	Constant
TEXT_VALUE	CONSTANT BINARY_INTEGER := 1
RAW_VALUE	CONSTANT BINARY_INTEGER := 2
BOOLEAN_VALUE	CONSTANT BINARY_INTEGER := 3

Table 132-1 (Cont.) DBMS_MGWMSG Constants: Value Types and Constants Representing the Type of Value for a SYS.MGW_NAME_VALUE_T Object

Value	Constant
BYTE_VALUE	CONSTANT BINARY_INTEGER := 4
SHORT_VALUE	CONSTANT BINARY_INTEGER := 5
INTEGER_VALUE	CONSTANT BINARY_INTEGER := 6
LONG_VALUE	CONSTANT BINARY_INTEGER := 7
FLOAT_VALUE	CONSTANT BINARY_INTEGER := 8
DOUBLE_VALUE	CONSTANT BINARY_INTEGER := 9
DATE_VALUE	CONSTANT BINARY_INTEGER := 10

Table 132-2 DBMS_MGWMSG Constants: Boolean Values—Constants Representing a Boolean as a Numeric Value

Value	Constant
BOOLEAN_FALSE	CONSTANT BINARY_INTEGER := 0
BOOLEAN_TRUE	CONSTANT BINARY_INTEGER := 1

Table 132-3 DBMS_MGWMSG Constants: Case Comparisons

Value	Constant
CASE_SENSITIVE	CONSTANT BINARY_INTEGER := 0
CASE_INSENSITIVE	CONSTANT BINARY_INTEGER := 1

Table 132-4 Constants for the TIB/Rendezvous field type

Value	Constant
TIBRVMSG_BOOL	CONSTANT INTEGER := 1
TIBRVMSG_F32	CONSTANT INTEGER := 2
TIBRVMSG_F64	CONSTANT INTEGER := 3
TIBRVMSG_I8	CONSTANT INTEGER := 4
TIBRVMSG_I16	CONSTANT INTEGER := 5
TIBRVMSG_I32	CONSTANT INTEGER := 6
TIBRVMSG_I64	CONSTANT INTEGER := 7
TIBRVMSG_IPADDR32	CONSTANT INTEGER := 8
TIBRVMSG_IPPORT16	CONSTANT INTEGER := 9
TIBRVMSG_DATETIME	CONSTANT INTEGER := 10
TIBRVMSG_F32ARRAY	CONSTANT INTEGER := 11
TIBRVMSG_F64ARRAY	CONSTANT INTEGER := 12
TIBRVMSG_I8ARRAY	CONSTANT INTEGER := 13

Table 132-4 (Cont.) Constants for the TIB/Rendezvous field type

Value	Constant
TIBRVMSG_I16ARRAY	CONSTANT INTEGER := 14
TIBRVMSG_I32ARRAY	CONSTANT INTEGER := 15
TIBRVMSG_I64ARRAY	CONSTANT INTEGER := 16
TIBRVMSG_OPAQUE	CONSTANT INTEGER := 17
TIBRVMSG_STRING	CONSTANT INTEGER := 18
TIBRVMSG_XML	CONSTANT INTEGER := 19

DBMS_MGWMSG Types

DBMS_MGWMSG uses types to specify certain information.

- [SYS.MGW_NAME_VALUE_T Type](#)
- [SYS.MGW_NAME_VALUE_T Type-Attribute Mapping](#)
- [SYS.MGW_NAME_TYPE_ARRAY_T Type](#)
- [SYS.MGW_TEXT_VALUE_T Type](#)
- [SYS.MGW_RAW_VALUE_T Type](#)
- [SYS.MGW_BASIC_MSG_T Type](#)
- [SYS.MGW_NUMBER_ARRAY_T Type](#)
- [SYS.MGW_TIBRV_FIELD_T Type](#)
- [SYS.MGW_TIBRV_MSG_T Type](#)

DBMS_MGWMSG SYS.MGW_NAME_VALUE_T Type

This type specifies a named value.

The name attribute, type attribute, and one of the <>_value attributes are typically not NULL.

Syntax

```
TYPE SYS.MGW_NAME_VALUE_T IS OBJECT(  
    name          VARCHAR2(250),  
    type          INTEGER,  
    integer_value  INTEGER,  
    number_value   NUMBER,  
    text_value     VARCHAR2(4000),  
    raw_value      RAW(2000),  
    date_value     DATE,  
  
    -- Methods  
    STATIC FUNCTION CONSTRUCT  
    RETURN SYS.MGW_NAME_VALUE_T,  
  
    STATIC FUNCTION CONSTRUCT_BOOLEAN (  
        name    IN VARCHAR2,  
        value   IN INTEGER )  
    RETURN SYS.MGW_NAME_VALUE_T,
```

```

STATIC FUNCTION CONSTRUCT_BYTE (
    name    IN VARCHAR2,
    value   IN INTEGER )
RETURN SYS.MGW_NAME_VALUE_T,

STATIC FUNCTION CONSTRUCT_SHORT (
    name    IN VARCHAR2,
    value   IN INTEGER )
RETURN SYS.MGW_NAME_VALUE_T,

STATIC FUNCTION CONSTRUCT_INTEGER (
    name    IN VARCHAR2,
    value   IN INTEGER )
RETURN SYS.MGW_NAME_VALUE_T,

STATIC FUNCTION CONSTRUCT_LONG (
    name    IN VARCHAR2,
    value   IN NUMBER )
RETURN SYS.MGW_NAME_VALUE_T,

STATIC FUNCTION CONSTRUCT_FLOAT (
    name    IN VARCHAR2,
    value   IN NUMBER )
RETURN SYS.MGW_NAME_VALUE_T,

STATIC FUNCTION CONSTRUCT_DOUBLE (
    name    IN VARCHAR2,
    value   IN NUMBER )
RETURN SYS.MGW_NAME_VALUE_T,

STATIC FUNCTION CONSTRUCT_TEXT (
    name    IN VARCHAR2,
    value   IN VARCHAR2 )
RETURN SYS.MGW_NAME_VALUE_T,

STATIC FUNCTION CONSTRUCT_RAW (
    name    IN VARCHAR2,
    value   IN RAW )
RETURN SYS.MGW_NAME_VALUE_T,

STATIC FUNCTION CONSTRUCT_DATE (
    name    IN VARCHAR2,
    value   IN DATE )
RETURN SYS.MGW_NAME_VALUE_T );

```

Attributes

Table 132-5 SYS.MGW_NAME_VALUE_T Attributes

Attribute	Description
name	Name associated with the value
type	Value type. Refer to the DBMS_MGWMSG.<>_VALUE constants in Table 132-1 . This indicates which Java datatype and class are associated with the value. It also indicates which attribute stores the value.
integer_value	Stores a numeric integer value
number_value	Stores a numeric float or large integer value

Table 132-5 (Cont.) SYS.MGW_NAME_VALUE_T Attributes

Attribute	Description
text_value	Stores a text value
raw_value	Stores a RAW (bytes) value
date_value	Stores a date value

SYS.MGW_NAME_VALUE_T Type-Attribute Mapping

[Table 132-6](#) shows the mapping between the value type and the attribute used to store the value.

Table 132-6 SYS.MGW_NAME_VALUE_T Type Attribute Mapping

Type	Value Stored in Attribute
DBMS_MGWMSG.TEXT_VALUE	text_value
DBMS_MGWMSG.RAW_VALUE	raw_value
DBMS_MGWMSG.BOOLEAN_VALUE	integer_value
DBMS_MGWMSG.BYTE_VALUE	integer_value
DBMS_MGWMSG.SHORT_VALUE	integer_value
DBMS_MGWMSG.INTEGER_VALUE	integer_value
DBMS_MGWMSG.LONG_VALUE	number_value
DBMS_MGWMSG.FLOAT_VALUE	number_value
DBMS_MGWMSG.DOUBLE_VALUE	number_value
DBMS_MGWMSG.DATE_VALUE	date_value

CONSTRUCT Method

This method constructs a new `SYS.MGW_NAME_VALUE_T` instance. All attributes are assigned a value of `NULL`.

Syntax

```
STATIC FUNCTION CONSTRUCT  
RETURN SYS.MGW_NAME_VALUE_T;
```

CONSTRUCT_TYPE Methods

These methods construct a new `SYS.MGW_NAME_VALUE_T` instance initialized with the value of a specific type. Each method sets the `name` and `type` attributes and one of the `<>_value` attributes, as shown in the mappings in [Table 132-6](#).

Syntax

```
STATIC FUNCTION CONSTRUCT_<> (  
    name    IN VARCHAR2,  
    value   IN datatype )  
RETURN SYS.MGW_NAME_VALUE_T;
```

Usage Notes

The `construct_boolean` method sets the value to either `DBMS_MGWMSG.BOOLEAN_TRUE` or `DBMS_MGWMSG.BOOLEAN_FALSE`.

DBMS_MGWMSG SYS.MGW_NAME_VALUE_ARRAY_T Type

This type specifies an array of name-value pairs. An object of `SYS.MGW_NAME_VALUE_ARRAY_T` type can have up to 1024 elements.

Syntax

```
TYPE SYS.MGW_NAME_VALUE_ARRAY_T
  AS VARRAY (1024) OF SYS.MGW_NAME_VALUE_T;
```

DBMS_MGWMSG SYS.MGW_TEXT_VALUE_T Type

This type specifies a `TEXT` value. It can store a large value as a `CLOB` or a smaller value (size ≤ 4000) as `VARCHAR2`. Only one of the `< >_value` attributes should be set.

Syntax

```
TYPE SYS.MGW_TEXT_VALUE_T IS OBJECT(
  small_value VARCHAR2(4000),
  large_value CLOB,

  -- Methods
  STATIC FUNCTION CONSTRUCT
  RETURN SYS.MGW_TEXT_VALUE_T);
```

Attributes

Table 132-7 SYS.MGW_TEXT_VALUE_T Attributes

Attribute	Description
<code>small_value</code>	Small <code>TEXT</code> value. Used for values ≤ 4000 .
<code>large_value</code>	Large <code>TEXT</code> value. Used when the value is too large for the <code>small_value</code> attribute.

CONSTRUCT Method

This method constructs a new `SYS.MGW_TEXT_VALUE_T` instance. All attributes are assigned a value of `NULL`.

Syntax

```
STATIC FUNCTION CONSTRUCT
  RETURN SYS.MGW_TEXT_VALUE_T;
```

DBMS_MGWMSG SYS.MGW_RAW_VALUE_T Type

This type specifies a `RAW` value. This type can store a large value as a `BLOB` or a smaller value (size ≤ 2000) as `RAW`. You must set no more than one of the `< >_value` attributes.

Syntax

```
TYPE SYS.MGW_RAW_VALUE_T IS OBJECT(
  small_value RAW(2000),
```

```
    large_value BLOB,  
  
--Methods  
STATIC FUNCTION CONSTRUCT  
RETURN SYS.MGW_RAW_VALUE_T);
```

Attributes

Table 132-8 SYS.MGW_RAW_VALUE_T Attributes

Attribute	Description
small_value	Small RAW (bytes) value <= 2000
large_value	Large RAW value. Used when the value is too large for the small_value attribute.

CONSTRUCT Method

This method constructs a new SYS.MGW_RAW_VALUE_T instance. All attributes are assigned a value of NULL.

Syntax

```
STATIC FUNCTION CONSTRUCT  
RETURN SYS.MGW_RAW_VALUE_T;
```

DBMS_MGWMSG SYS.MGW_BASIC_MSG_T Type

This is a canonical type for a basic TEXT or RAW message. Only a single TEXT or RAW value is typically set. An object of this type must not have both TEXT and RAW set to a not NULL value at the same time.

Syntax

```
TYPE SYS.MGW_BASIC_MSG_T IS OBJECT(  
    header      SYS.MGW_NAME_VALUE_ARRAY_T,  
    text_body   SYS.MGW_TEXT_VALUE_T,  
    raw_body    SYS.MGW_RAW_VALUE_T,  
  
--Methods  
STATIC FUNCTION CONSTRUCT  
RETURN SYS.MGW_BASIC_MSG_T);
```

Attributes

Table 132-9 SYS.MGW_BASIC_MSG_T Attributes

Attribute	Description
header	Message header information as an array of name-value pairs
text_body	Message body for a TEXT message
raw_body	Message body for a RAW (bytes) message

CONSTRUCT Method

This method constructs a new SYS.MGW_BASIC_MSG_T instance. All attributes are assigned a value of NULL.

Syntax

```

STATIC FUNCTION CONSTRUCT
RETURN SYS.MGW_BASIC_MSG_T;

```

DBMS_MGWMSG SYS.MGW_NUMBER_ARRAY_T Type

A type that specifies an array of numbers.

Syntax

```

TYPE SYS.MGW_NUMBER_ARRAY_T AS VARRAY(1024) OF NUMBER;

```

DBMS_MGWMSG SYS.MGW_TIBRV_FIELD_T Type

This type represents a TIB/Rendezvous message field, typically used in a read-only fashion to retrieve field information from a SYS.MGW_TIBRV_MSG_T instance.

Syntax

```

TYPE SYS.MGW_TIBRV_FIELD_T IS OBJECT(
    field_name      VARCHAR2(256),
    field_id        INTEGER,
    field_type      INTEGER,
    number_value    NUMBER,
    number_array_value SYS.MGW_NUMBER_ARRAY_T,
    text_value      VARCHAR2(4000),
    raw_value       RAW(2000),
    date_value      DATE,
    clob_value      CLOB,
    blob_value      BLOB);

```

Attributes

Table 132-10 SYS.MGW_TIBRV_FIELD_T Attributes

Attribute	Description
field_name	Field name. This will be NULL if the field has no name.
field_id	Field identifier. If the field identifier is zero (0), then that field is considered not to have a field identifier. Otherwise the field identifier is a nonzero value that is unique for all fields of that message.
field_type	Field wire format datatype. The DBMS_MGWMSG.TIBRVMSG_<> constants represent valid values for this attribute. The value of this field discriminates which value attribute is used to store the field data.
number_value	Used to store a numeric value
number_array_value	Used to store a numeric array value
text_value	Used to store a small text value
raw_value	Used to store a small raw value
date_value	Used to store a date value
clob_value	Used to store a large text value. This is used when the text data will not fit in text_value, that is, when size is larger than 4000.
blob_value	Used to store a large raw value. This is used when the raw data will not fit in raw_value; that is, when size is larger than 2000.

SYS.MGW_TIBRV_FIELD_T Type and Attribute Mapping

Table 132-11 describes the mapping in type `SYS.MGW_TIBRV_FIELD_T` between the field type and attribute used to store the value.

Table 132-11 SYS.MGW_TIBRV_FIELD_T Type and Attribute Mapping

Field Type (DBMS_MGWMSG constant)	Value Stored in Attribute
<code>TIBRVMSG_BOOL</code>	<code>number_value</code>
<code>TIBRVMSG_F32</code>	<code>number_value</code>
<code>TIBRVMSG_F64</code>	<code>number_value</code>
<code>TIBRVMSG_I8</code>	<code>number_value</code>
<code>TIBRVMSG_I16</code>	<code>number_value</code>
<code>TIBRVMSG_I32</code>	<code>number_value</code>
<code>TIBRVMSG_I64</code>	<code>number_value</code>
<code>TIBRVMSG_IPADDR32</code>	<code>text_value</code>
<code>TIBRVMSG_IPPORT16</code>	<code>number_value</code>
<code>TIBRVMSG_DATETIME</code>	<code>date_value</code>
<code>TIBRVMSG_F32ARRAY</code>	<code>number_array_value</code>
<code>TIBRVMSG_F64ARRAY</code>	<code>number_array_value</code>
<code>TIBRVMSG_I8ARRAY</code>	<code>number_array_value</code>
<code>TIBRVMSG_I16ARRAY</code>	<code>number_array_value</code>
<code>TIBRVMSG_I32ARRAY</code>	<code>number_array_value</code>
<code>TIBRVMSG_I64ARRAY</code>	<code>number_array_value</code>
<code>TIBRVMSG_OPAQUE</code>	<code>raw_value</code> or <code>blob_value</code>
<code>TIBRVMSG_STRING</code>	<code>text_value</code> or <code>clob_value</code>
<code>TIBRVMSG_XML</code>	<code>raw_value</code> or <code>blob_value</code>

DBMS_MGWMSG SYS.MGW_TIBRV_MSG_T Type

This type represents a TIB/Rendezvous message. You must never directly reference the attributes of this type. Instead use the type methods.

Syntax

```
TYPE SYS.MGW_TIBRV_MSG_T IS OBJECT(
    send_subject    VARCHAR2(256),
    reply_subject   VARCHAR2(256),
    cm_time_limit   NUMBER,
    cm_sender_name  VARCHAR2(256),
    cm_sequence_num NUMBER,
    fields          SYS.MGW_TIBRV_IFIELDS_T,
    clob_data1      CLOB,
    clob_data2      CLOB,
    clob_data3      CLOB,
    blob_data1      BLOB,
    blob_data2      BLOB,
```

```
        blob_data3      BLOB,

    STATIC FUNCTION construct
    RETURN SYS.MGW_TIBRV_MSG_T,

    MEMBER PROCEDURE add_bool (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  INTEGER ),

    MEMBER PROCEDURE add_f32 (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  FLOAT ),

    MEMBER PROCEDURE add_f64 (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  DOUBLE ),

    MEMBER PROCEDURE add_i8 (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  INTEGER ),

    MEMBER PROCEDURE add_i16 (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  INTEGER ),

    MEMBER PROCEDURE add_i32 (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  INTEGER ),

    MEMBER PROCEDURE add_i64 (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  NUMBER ),

    MEMBER PROCEDURE add_ipaddr32 (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  VARCHAR2 ),

    MEMBER PROCEDURE add_ipport16 (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  INTEGER ),

    MEMBER PROCEDURE add_datetime (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  DATE ),

    MEMBER PROCEDURE add_f32array (
        name IN  VARCHAR2,
        id   IN   INTEGER,
        value IN  SYS.MGW_NUMBER_ARRAY_T ),

    MEMBER PROCEDURE add_f64array (
        name IN  VARCHAR2,
```

```
        id    IN  INTEGER,
        value IN  SYS.MGW_NUMBER_ARRAY_T ),

MEMBER PROCEDURE add_i8array (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  SYS.MGW_NUMBER_ARRAY_T ),

MEMBER PROCEDURE add_i16array (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  SYS.MGW_NUMBER_ARRAY_T ),

MEMBER PROCEDURE add_i32array (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  SYS.MGW_NUMBER_ARRAY_T ),

MEMBER PROCEDURE add_i64array (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  SYS.MGW_NUMBER_ARRAY_T ),

MEMBER PROCEDURE add_string (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  VARCHAR2 ),

MEMBER PROCEDURE add_string (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  CLOB ),

MEMBER PROCEDURE add_opaque (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  RAW ),

MEMBER PROCEDURE add_opaque (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  BLOB ),

MEMBER PROCEDURE add_xml (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  RAW ),

MEMBER PROCEDURE add_xml (
    name IN  VARCHAR2,
    id    IN  INTEGER,
    value IN  BLOB ),

MEMBER PROCEDURE set_send_subject (
    value IN  VARCHAR2 ),

MEMBER PROCEDURE set_reply_subject (
    value IN  VARCHAR2 ),

MEMBER PROCEDURE set_cm_time_limit (
    value IN  NUMBER ),
```

```
MEMBER PROCEDURE set_cm_sender_name (  
    value IN VARCHAR2 ),  
  
MEMBER PROCEDURE set_cm_sequence_num (  
    value IN NUMBER ),  
  
MEMBER FUNCTION get_send_subject  
RETURN VARCHAR2,  
  
MEMBER FUNCTION get_reply_subject  
RETURN VARCHAR2,  
  
MEMBER FUNCTION get_cm_time_limit  
RETURN NUMBER,  
  
MEMBER FUNCTION get_cm_sender_name  
RETURN VARCHAR2,  
  
MEMBER FUNCTION get_cm_sequence_num  
RETURN NUMBER,  
  
MEMBER FUNCTION get_field_count  
RETURN INTEGER,  
  
MEMBER FUNCTION get_field (  
    idx    IN INTEGER )  
RETURN SYS.MGW_TIBRV_FIELD_T,  
  
MEMBER FUNCTION get_field_by_name (  
    name   IN VARCHAR2 )  
RETURN SYS.MGW_TIBRV_FIELD_T,  
  
MEMBER FUNCTION get_field_by_id (  
    id     IN INTEGER )  
RETURN SYS.MGW_TIBRV_FIELD_T,  
  
MEMBER FUNCTION find_field_name (  
    name   IN VARCHAR2,  
    start_idx IN INTEGER )  
RETURN INTEGER,  
  
MEMBER FUNCTION find_field_id (  
    id     IN INTEGER,  
    start_idx IN INTEGER )  
RETURN INTEGER  
);
```

Attributes

Table 132-12 SYS.MGW_TIBRV_MSG_T Type Attributes

Attribute	Description
send_subject	Send subject name
reply_subject	Reply subject name
cm_time_limit	Time limit for a certified message
cm_sender_name	Sender name of a certified message
cm_sequence_num	Sequence number of a certified message

Table 132-12 (Cont.) SYS.MGW_TIBRV_MSG_T Type Attributes

Attribute	Description
fields	Collection of message fields
clob_data1	Used to store a large text value
clob_data2	Used to store a large text value
clob_data3	Used to store a large text value
blob_data1	Used to store a large raw value
blob_data2	Used to store a large raw value
blob_data3	Used to store a large raw value

Construct Method

Constructs a new SYS.MGW_TIBRV_MSG_T instance. All attributes are set to NULL.

Syntax

```
STATIC FUNCTION construct
RETURN SYS.MGW_TIBRV_MSG_T;
```

ADD_<> Methods

Adds a new field to the message.

Syntax

```
MEMBER PROCEDURE ADD_<> (
    name IN VARCHAR2,
    id IN INTEGER,
    value IN datatype );
```

Parameters**Table 132-13 SYS.MGW_TIBRV_MSG_T ADD_<> Method Parameters**

Parameter	Description
name	Field name
id	Field identifier
value	Field data

[Table 132-14](#) shows, for each add method, the field type that will be assigned and valid values for the field data.

Table 132-14 MGW_TIBRV_MSG_T Add Method Field Types

Method Name	Field Type Assigned	Comment
add_bool	TIBRVMSG_BOOL	Valid values: 0 (false), 1 (true)
add_f32	TIBRVMSG_F32	n/a
add_f64	TIBRVMSG_F64	n/a
add_i8	TIBRVMSG_I8	Valid range: -128...127

Table 132-14 (Cont.) MGW_TIBRV_MSG_T Add Method Field Types

Method Name	Field Type Assigned	Comment
add_i16	TIBRVMSG_I16	Valid range: -32768...32767
add_i32	TIBRVMSG_I32	Valid range: -2147483648... 2147483647
add_i64	TIBRVMSG_I64	n/a
add_ipaddr32	TIBRVMSG_IPADDR32	n/a
add_ipport16	TIBRVMSG_IPPORT16	n/a
add_datetime	TIBRVMSG_DATETIME	n/a
add_f32array	TIBRVMSG_F32ARRAY	n/a
add_f64array	TIBRVMSG_F64ARRAY	n/a
add_i8array	TIBRVMSG_I8ARRAY	Valid range: -128...127
add_i16array	TIBRVMSG_I16ARRAY	Valid range: -32768...32767
add_i32array	TIBRVMSG_I32ARRAY	Valid range: -2147483648... 2147483647
add_i64array	TIBRVMSG_I64ARRAY	n/a
add_opaque	TIBRVMSG_OPAQUE	Value stored as RAW if size < 2000; otherwise value stored in BLOB
add_string	TIBRVMSG_STRING	Value stored as VARCHAR2 if size < 4000; otherwise value stored in CLOB
add_xml	TIBRVMSG_XML	Value stored as RAW if size < 2000; otherwise value stored in BLOB

SET_<> Methods

Accessor methods to set an instance attribute to a specific value.

Syntax

```
MEMBER PROCEDURE SET_<> (
    value IN datatype );
```

Parameters**Table 132-15 SYS.MGW_TIBRV_MSG_T SET_<> Method Parameters**

Parameter	Description
value	Value to be assigned

GET_<> Methods

Accessor methods to retrieve the value for an instance attribute.

Syntax

```
MEMBER PROCEDURE GET_<>
RETURN datatype;
```

Parameters

None

Return Values

Returns the attribute value.

GET_FIELD_COUNT Function

Gets the number of message fields.

Syntax

```
MEMBER PROCEDURE get_field_count  
RETURN INTEGER;
```

Parameters

None

Return Values

Returns the number of fields, or zero (0) if there are none.

GET_FIELD Function

Retrieves field information for the field having a given field collection index. This method should only be called if the `GET_FIELD_COUNT` Function returns a nonzero value and `idx` must specify a valid collection index; that is, `1<=idx<=get_field_count()`.

Syntax

```
MEMBER PROCEDURE get_field (  
    idx    IN    INTEGER )  
RETURN SYS.MGW_TIBRV_FIELD_T;
```

Parameters

Table 132-16 SYS.MGW_TIBRV_MSG_T GET_FIELD Function Parameters

Parameter	Description
<code>idx</code>	Specifies the 1-based field collection index of the field to retrieve



Note:

A 1-based index begins at one (1) instead of zero (0).

Return Values

Returns the field information.

GET_FIELD_BY_NAME Function

Retrieves field information for the first field that has a given field name. The name comparison is case-sensitive.

Syntax

```
MEMBER PROCEDURE get_field_by_name (  
    name IN VARCHAR2 )  
RETURN SYS.MGW_TIBRV_FIELD_T;
```

Parameters

Table 132-17 SYS.MGW_TIBRV_MSG_T GET_FIELD_BY_NAME Function Parameters

Parameter	Description
name	Specifies the field name to search for. This can be <code>NULL</code> to find the first field that does not have a field name.

Return Values

Returns the field information, or `NULL` if no match was found.

GET_FIELD_BY_ID Function

Retrieves field information for the first field that has a given field identifier.

A field can have either a unique identifier or no identifier. If the field identifier value is zero (0) or `NULL`, then the field is considered to have no identifier. Otherwise, the identifier is a nonzero value that is unique for all the fields of this message.

Syntax

```
MEMBER PROCEDURE get_field_by_id (  
    id IN INTEGER )  
RETURN SYS.MGW_TIBRV_FIELD_T;
```

Parameters

Table 132-18 SYS.MGW_TIBRV_MSG_T GET_FIELD_BY_ID Function Parameters

Parameter	Description
id	Specifies the field identifier to search for. This can be zero (0) or <code>NULL</code> to find the first field that does not have an identifier.

Return Values

Returns the field information, or `NULL` if no match was found.

FIND_FIELD_NAME Function

Searches for a field with a given field name, starting from a given index of the field collection. It returns the index of that field. The name comparison is case-sensitive. This function is useful for finding all the fields that have the same name.

Syntax

```
MEMBER PROCEDURE find_field_name (  
    name IN VARCHAR2,  
    start_idx IN INTEGER )  
RETURN INTEGER;
```

Parameters

Table 132-19 SYS.MGW_TIBRV_MSG_T FIND_FIELD_NAME Function Parameters

Parameter	Description
name	Specifies the field name to search for. This can be NULL to search for a field that does not have a field name.
start_idx	Specifies the 1-based field collection index from which the search should start.

Return Values

Returns the field index (> 0) if a match was found, or zero (0) if no match was found.

FIND_FIELD_ID Function

Searches for a field with a given field identifier, starting from a given index of the field collection. It returns the index of that field.

Syntax

```
MEMBER PROCEDURE find_field_id (  
    id          IN  INTEGER,  
    start_idx IN INTEGER )  
RETURN        INTEGER;
```

Parameters**Table 132-20 SYS.MGW_TIBRV_MSG_T FIND_FIELD_ID Function Parameters**

Parameter	Description
id	Specifies the field identifier to search for. This can be zero (0) or NULL to find a field that does not have an identifier.
start_idx	Specifies the 1-based field collection index from which the search should start.

Return Values

Returns the field index (> 0) if a match was found, or zero (0) if no match was found.

Summary of DBMS_MGWMSG Subprograms

This table lists the DBMS_MGWMSG subprograms and briefly describes them.

Table 132-21 DBMS_MGWMSG Package Subprograms

Subprogram	Description
LCR_TO_XML Function	Converts a SYS.ANYDATA object encapsulating a row LCR (LCR\$_ROW_RECORD) or a DDL LCR (LCR\$_DDL_RECORD) to a SYS.XMLTYPE object
NVARRAY_ADD Procedure	Appends a name-value element to the end of a name-value array
NVARRAY_FIND_NAME Function	Searches a name-value array for the element with the name you specify in p_name

Table 132-21 (Cont.) DBMS_MGWMSG Package Subprograms

Subprogram	Description
NVARARRAY_FIND_NAME_TYPE Function	Searches a name-value array for an element with the name and value type you specify
NVARARRAY_GET Function	Gets the name-value element of the name you specify in <code>p_name</code> from a name-value array
NVARARRAY_GET_BOOLEAN Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>BOOLEAN_VALUE</code> value type
NVARARRAY_GET_BYTE Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>BYTE_VALUE</code> value type
NVARARRAY_GET_DATE Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>DATE_VALUE</code> value type
NVARARRAY_GET_DOUBLE Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>DOUBLE_VALUE</code> value type
NVARARRAY_GET_FLOAT Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>FLOAT_VALUE</code> value type
NVARARRAY_GET_INTEGER Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>INTEGER_VALUE</code> value type
NVARARRAY_GET_LONG Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>LONG_VALUE</code> value type
NVARARRAY_GET_RAW Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>RAW_VALUE</code> value type
NVARARRAY_GET_SHORT Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>SHORT_VALUE</code> value type
NVARARRAY_GET_TEXT Function	Gets the value of the name-value array element that you specify in <code>p_name</code> and with the <code>TEXT_VALUE</code> value type
XML_TO_LCR Function	Converts a <code>SYS.XMLTYPE</code> object to a <code>SYS.ANYDATA</code> object encapsulating a row LCR (<code>LCR\$_ROW_RECORD</code>) or a DDL LCR (<code>LCR\$_DDL_RECORD</code>)

LCR_TO_XML Function

This function converts a `SYS.ANYDATA` object encapsulating a row LCR (Logical Change Record, in this case a `LCR$_ROW_RECORD`) or a DDL LCR (`LCR$_DDL_RECORD`) to a `SYS.XMLTYPE` object.



See Also:

[XML_TO_LCR Function](#)

Syntax

```
DBMS_MGWMSG.LCR_TO_XML (
    p_anydata IN SYS.ANYDATA )
RETURN SYS.XMLTYPE;
```

Parameters

Table 132-22 LCR_TO_XML Function Parameters

Parameter	Description
p_anydata	An ANYDATA object to be converted

Return Values

Returns a SYS.XMLTYPE object.

Usage Notes

An exception is raised if the encapsulated type p_anydata is not an LCR.

NVARRAY_ADD Procedure

This procedure appends a name-value element to the end of a name-value array.

Syntax

```
DBMS_MGWMSG.NVARRAY_ADD (  
    p_array IN OUT SYS.MGW_NAME_VALUE_ARRAY_T,  
    p_value IN      SYS.MGW_NAME_VALUE_T );
```

Parameters

Table 132-23 NVARRAY_ADD Procedure Parameters

Parameter	Description
p_array	On input, the name-value array instance to modify. If NULL, then a new array is created. On output, the modified name-value array instance.
p_value	The value to add. If NULL, then p_array is not changed.

NVARRAY_FIND_NAME Function

This function searches a name-value array for the element with the name you specify in p_name.

Syntax

```
DBMS_MGWMSG.NVARRAY_FIND_NAME (  
    p_array IN SYS.MGW_NAME_VALUE_ARRAY_T,  
    p_name IN VARCHAR2,  
    p_compare IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )  
RETURN BINARY_INTEGER;
```

Parameters

Table 132-24 NVARRAY_FIND_NAME Function Parameters

Parameters	Description
p_array	The name-value array to search
p_name	The name to find
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns a positive integer that is the array index of the matching element or zero (0) if the specified name is not found.

NVARRAY_FIND_NAME_TYPE Function

This function searches a name-value array for an element with the name and value type you specify.

Syntax

```
DBMS_MGWMSG.NVARRAY_FIND_NAME_TYPE (  
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,  
    p_name     IN VARCHAR2,  
    p_type     IN BINARY_INTEGER  
    p_compare  IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )  
RETURN BINARY_INTEGER;
```

Parameters

Table 132-25 NVARRAY_FIND_NAME_TYPE Function Parameters

Parameter	Description
p_array	The name-value array to search
p_name	The name to find
p_type	The value type. Refer to the value type constants in Table 132-1 .
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns a positive integer that is the array index of the matching element, zero (0) if the specified name is not found, or negative one (-1) if the specified name is found but a type mismatch exists.

NVARRAY_GET Function

This function gets the name-value element of the name you specify in `p_name` from a name-value array.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET (  
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,  
    p_name     IN VARCHAR2,  
    p_compare  IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )  
RETURN SYS.MGW_NAME_VALUE_T;
```

Parameters

Table 132-26 NVARRAY_GET Function Parameters

Parameter	Description
<code>p_array</code>	The name-value array
<code>p_name</code>	The value name
<code>p_compare</code>	Name comparison method. Values are <code>CASE_SENSITIVE</code> and <code>CASE_INSENSITIVE</code> .

Return Values

Returns the matching element, or `NULL` if the specified name is not found.

NVARRAY_GET_BOOLEAN Function

This function gets the value of the name-value array element that you specify in `p_name` and with the `BOOLEAN_VALUE` value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_BOOLEAN (  
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,  
    p_name     IN VARCHAR2,  
    p_compare  IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )  
RETURN INTEGER;
```

Parameters

Table 132-27 NVARRAY_GET_BOOLEAN Function Parameters

Parameter	Description
<code>p_array</code>	The name-value array
<code>p_name</code>	The value name
<code>p_compare</code>	Name comparison method. Values are <code>CASE_SENSITIVE</code> and <code>CASE_INSENSITIVE</code> .

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_BYTE Function

This function gets the value of the name-value array element that you specify in `p_name` and with the `BYTE_VALUE` value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_BYTE (
    p_array      IN SYS.MGW_NAME_VALUE_ARRAY_T,
    p_name       IN VARCHAR2,
    p_compare    IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )
RETURN INTEGER;
```

Parameters

Table 132-28 NVARRAY_GET_BYTE Function Parameters

Parameter	Description
p_array	The name-value array
p_name	The value name
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_DATE Function

This function gets the value of the name-value array element that you specify in `p_name` and with the `DATE_VALUE` value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_DATE (
    p_array      IN SYS.MGW_NAME_VALUE_ARRAY_T,
    p_name       IN VARCHAR2,
    p_compare    IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )
RETURN DATE;
```

Parameters

Table 132-29 NVARRAY_GET_DATE Function Parameters

Parameters	Description
p_array	The name-value array
p_name	The value name

Table 132-29 (Cont.) NVARRAY_GET_DATE Function Parameters

Parameters	Description
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_DOUBLE Function

This function gets the value of the name-value array element that you specify in p_name and with the DOUBLE_VALUE value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_DOUBLE (
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,
    p_name     IN VARCHAR2,
    p_compare  IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )
RETURN NUMBER;
```

Parameters

Table 132-30 NVARRAY_GET_DOUBLE Function Parameters

Parameter	Description
p_array	The name-value array
p_name	The value name
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_FLOAT Function

This function gets the value of the name-value array element that you specify in p_name and with the FLOAT_VALUE value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_FLOAT (
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,
    p_name     IN VARCHAR2,
    p_compare  IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )
RETURN NUMBER;
```

Parameters

Table 132-31 NVARRAY_GET_FLOAT Function Parameters

Parameter	Description
p_array	The name-value array
p_name	The value name
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_INTEGER Function

This function gets the value of the name-value array element that you specify in p_name and with the INTEGER_VALUE value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_INTEGER (
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,
    p_name     IN VARCHAR2,
    p_compare  IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )
RETURN INTEGER;
```

Parameters

Table 132-32 NVARRAY_GET_INTEGER Function Parameters

Parameter	Description
p_array	The name-value array
p_name	The value name
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_LONG Function

This function gets the value of the name-value array element that you specify in p_name and with the LONG_VALUE value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_LONG (
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,
    p_name     IN VARCHAR2,
```



```
    p_compare IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )  
RETURN NUMBER;
```

Parameters

Table 132-33 NVARRAY_GET_LONG Function Parameters

Parameter	Description
p_array	The name-value array
p_name	The value name
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_RAW Function

This function gets the value of the name-value array element that you specify in p_name and with the RAW_VALUE value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_RAW (  
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,  
    p_name     IN VARCHAR2,  
    p_compare  IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )  
RETURN RAW;
```

Parameters

Table 132-34 NVARRAY_GET_RAW Function Parameters

Parameter	Description
p_array	The name-value array
p_name	The value name
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_SHORT Function

This function gets the value of the name-value array element that you specify in p_name and with the SHORT_VALUE value type.

Syntax

```
DBMS_MGWMSG.NVARRAY_GET_SHORT (  
    p_array    IN SYS.MGW_NAME_VALUE_ARRAY_T,
```

```

    p_name      IN VARCHAR2,
    p_compare   IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )
RETURN INTEGER;
```

Parameters

Table 132-35 NVARRAY_GET_SHORT Function Parameters

Parameter	Description
p_array	The name-value array
p_name	The value name
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

NVARRAY_GET_TEXT Function

This function gets the value of the name-value array element that you specify in p_name and with the TEXT_VALUE value type.

Syntax

```

DBMS_MGWMSG.NVARRAY_GET_TEXT (
    p_array      IN SYS.MGW_NAME_VALUE_ARRAY_T,
    p_name       IN VARCHAR2,
    p_compare    IN BINARY_INTEGER DEFAULT CASE_SENSITIVE )
RETURN VARCHAR2;
```

Parameters

Table 132-36 NVARRAY_GET_TEXT Function Parameters

Parameter	Description
p_array	The name-value array
p_name	The value name
p_compare	Name comparison method. Values are CASE_SENSITIVE and CASE_INSENSITIVE.

Return Values

Returns the value, or NULL if either the specified name is not found or a type mismatch exists.

XML_TO_LCR Function

This function converts a `SYS.XMLTYPE` object to a `SYS.ANYDATA` object encapsulating a row LCR (`LCR$_ROW_RECORD`) or a DDL LCR (`LCR$_DDL_RECORD`).



See Also:

[LCR_TO_XML Function](#)

Syntax

```
DBMS_MGWMSG.XML_TO_LCR (
    p_xmldata IN SYS.XMLTYPE )
RETURN SYS.ANYDATA;
```

Parameters

Table 132-37 XML_TO_LCR Function Parameters

Parameter	Description
p_xmldata	An XMLTYPE object representing an LCR

Return Values

Returns a `SYS.ANYDATA` object.

Usage Notes

An exception is raised if `p_xmldata` cannot be converted to an LCR.