

---

**[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)**[PREV CLASS](#) [NEXT CLASS](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)SUMMARY: NESTED | [ENUM CONSTANTS](#) | FIELD | [METHOD](#)DETAIL: [ENUM CONSTANTS](#) | FIELD | [METHOD](#)

---

javax.ejb

## Enum TransactionAttributeType

java.lang.Object

└─ java.lang.Enum<[TransactionAttributeType](#)>

└─ javax.ejb.TransactionAttributeType

### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<[TransactionAttributeType](#)>

```
public enum TransactionAttributeType
extends java.lang.Enum<TransactionAttributeType>
```

The enum `TransactionAttributeType` is used with the `TransactionAttribute` annotation to specify whether the methods of a session bean or message driven bean are called with a valid transaction context.

For a message-driven bean's message listener methods (or interface), only the `REQUIRED` and `NOT_SUPPORTED` values may be used.

For an enterprise bean's timeout callback methods, only the `REQUIRED`, `REQUIRES_NEW` and `NOT_SUPPORTED` values may be used.

For a session bean's asynchronous business methods, only the `REQUIRED`, `REQUIRES_NEW`, and `NOT_SUPPORTED` values may be used.

For a singleton session bean's `PostConstruct` and `PreDestroy` lifecycle callback interceptor methods, only the `REQUIRED`, `REQUIRES_NEW`, and `NOT_SUPPORTED` values may be used.

If an enterprise bean implements the `SessionSynchronization` interface or uses any of the session synchronization annotations, only the following values may be used for the transaction attributes of the bean's methods: `REQUIRED`, `REQUIRES_NEW`, `MANDATORY`.

### Since:

EJB 3.0

### See Also:

`TransactionAttribute`

---

## Enum Constant Summary

### [MANDATORY](#)

If a client invokes the enterprise bean's method while the client is associated with a transaction context, the container invokes the enterprise bean's method in the client's transaction context.

### [NEVER](#)

The client is required to call without a transaction context, otherwise an exception is thrown.

### [NOT\\_SUPPORTED](#)

The container invokes an enterprise bean method whose transaction attribute `NOT_SUPPORTED` with an unspecified transaction context.

### [REQUIRED](#)

If a client invokes the enterprise bean's method while the client is associated with a transaction context, the container invokes the enterprise bean's method in the client's

transaction context.

#### [REQUIRES\\_NEW](#)

The container must invoke an enterprise bean method whose transaction attribute is set to `REQUIRES_NEW` with a new transaction context.

#### [SUPPORTS](#)

If the client calls with a transaction context, the container performs the same steps as described in the `REQUIRED` case.

## Method Summary

static <a href="#">TransactionAttributeType</a>	<a href="#">valueOf</a> (java.lang.String name) Returns the enum constant of this type with the specified name.
static <a href="#">TransactionAttributeType</a> []	<a href="#">values</a> () Returns an array containing the constants of this enum type, in the order they are declared.

### Methods inherited from class java.lang.Enum

`clone`, `compareTo`, `equals`, `finalize`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

### Methods inherited from class java.lang.Object

`getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Enum Constant Detail

### MANDATORY

```
public static final TransactionAttributeType MANDATORY
```

If a client invokes the enterprise bean's method while the client is associated with a transaction context, the container invokes the enterprise bean's method in the client's transaction context.

If there is no existing transaction, an exception is thrown.

### REQUIRED

```
public static final TransactionAttributeType REQUIRED
```

If a client invokes the enterprise bean's method while the client is associated with a transaction context, the container invokes the enterprise bean's method in the client's transaction context.

If the client invokes the enterprise bean's method while the client is not associated with a transaction context, the container automatically starts a new transaction before delegating a method call to the enterprise bean method.

### REQUIRES\_NEW

```
public static final TransactionAttributeType REQUIRES_NEW
```

The container must invoke an enterprise bean method whose transaction attribute is set to `REQUIRES_NEW` with a new transaction context.

If the client invokes the enterprise bean's method while the client is not associated with a transaction context, the container automatically starts a new transaction before delegating a method call to the enterprise bean business method.

If a client calls with a transaction context, the container suspends the association of the transaction context with the current thread before starting the new transaction and invoking the method. The container resumes the suspended transaction association after the method and the new transaction have been completed.

---

## SUPPORTS

```
public static final TransactionAttributeType SUPPORTS
```

If the client calls with a transaction context, the container performs the same steps as described in the `REQUIRED` case.

If the client calls without a transaction context, the container performs the same steps as described in the `NOT_SUPPORTED` case.

The `SUPPORTS` transaction attribute must be used with caution. This is because of the different transactional semantics provided by the two possible modes of execution. Only enterprise beans that will execute correctly in both modes should use the `SUPPORTS` transaction attribute.

---

## NOT\_SUPPORTED

```
public static final TransactionAttributeType NOT_SUPPORTED
```

The container invokes an enterprise bean method whose transaction attribute `NOT_SUPPORTED` with an unspecified transaction context.

If a client calls with a transaction context, the container suspends the association of the transaction context with the current thread before invoking the enterprise bean's business method. The container resumes the suspended association when the business method has completed.

---

## NEVER

```
public static final TransactionAttributeType NEVER
```

The client is required to call without a transaction context, otherwise an exception is thrown.

## Method Detail

### values

```
public static TransactionAttributeType[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (TransactionAttributeType c : TransactionAttributeType.values())  
    System.out.println(c);
```

### Returns:

an array containing the constants of this enum type, in the order they are declared

## valueOf

```
public static TransactionAttributeType valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

### Parameters:

name - the name of the enum constant to be returned.

### Returns:

the enum constant with the specified name

### Throws:

IllegalArgumentException - if this enum type has no constant with the specified name

java.lang.NullPointerException - if the argument is null

---

## [Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [ENUM CONSTANTS](#) | [FIELD](#) | [METHOD](#)

DETAIL: [ENUM CONSTANTS](#) | [FIELD](#) | [METHOD](#)

---

[Submit a bug or feature](#)

Copyright © 2009-2011, Oracle Corporation and/or its affiliates. All Rights Reserved. Use is subject to [license terms](#).

Generated on 10-February-2011 12:41