173

DBMS_SAGA

The `DBMS_SAGA` package provides a collection of saga functions and procedures to initiate and finalize sagas.

This chapter contains the following topics:

DBMS_SAGA Overview

The `DBMS_SAGA` package enables you (developers) to use PL/SQL and develop packaged microservices applications in the database without requiring a mid-tier.

The `DBMS_SAGA` package provides the PL/SQL interfaces to allow client programs to interact with database sagas.

If you want to implement microservices using a mid-tier to communicate with a database, it is recommended that you use the AQJMS extensions to implement the saga functionality.

See Also:

Managing a Saga Using JMS Interface for more information about the AQJMS extensions.

DBMS_SAGA Security Model

The `DBMS_SAGA` package requires the `SAGA_PARTICPANT` role to participate in sagas.

Summary of DBMS_SAGA Subprograms

This table lists and briefly describes the DBMS_SAGA package subprograms.

Table 173-1 DBMS_SAGA Package Subprograms

Subprogram	Description
BEGIN_SAGA Function	Creates and returns a new `saga_id_t` (GUID). The new Saga ID is inserted into the `Saga$` dictionary table.
GET_SAGA_ID Function	Gets the Saga ID for a Saga associated with a database session.
SET_SAGA_ID Procedure	Sets the Saga ID for a database session to the `saga_id` provided.
COMMIT_SAGA Procedure	Commits the Saga identified by the `saga_id` parameter.

Table 173-1 (Cont.) DBMS_SAGA Package Subprograms

Subprogram	Description
ROLLBACK_SAGA Procedure	Rolls back the Saga identified by the `saga_id` parameter.
IS_INCOMPLETE Function	Returns `TRUE` if the Saga corresponding to the provided `saga_id` is considered incomplete.
SET_INCOMPLETE Procedure	Marks the Saga identified by the provided `saga_id` as incomplete.
SEND_REQUEST Procedure	Enrolls a participant to an initiated Saga.

BEGIN_SAGA Function

This function creates and returns a new saga GUID: `saga_id_t`. The new saga ID is inserted into the `saga$` dictionary table.

Syntax

```
function begin_saga(initiator_name IN varchar2,
          timeout   IN number default NULL,
          version   IN number default 1)
 return saga_id_t;
```

Parameters

Table 173-2 BEGIN_SAGA Procedure Parameters

Parameter	Description
`initiator_name`	The name of the saga initiator.
`timeout`	The timeout value specified as number of seconds. This value overrides the value of the `max_saga_duration` parameter.

GET_SAGA_ID Function

This function returns the saga, if any, associated with the current database session.

Syntax

```
function get_saga_id() return saga_id_t;
```

SET_SAGA_ID Procedure

This procedure sets the saga ID for a database session to the `saga_id` provided.

Syntax

```
procedure set_saga_id(saga_id  IN saga_id_t);
```

Parameters

Table 173-3 SET_SAGA Procedure Parameters

Parameter	Description
`saga_id`	The saga identifier of the type `saga_id_t`.

COMMIT_SAGA Procedure

This procedure commits the saga that is identified with the `saga_id` parameter.

The `force` option is used to forcefully commit a saga participant transaction for pending or incomplete sagas, For the `COMMIT_SAGA()` call on the participant database, `force=TRUE` only commits the saga branch operation on the participant database. The default value of `force` is `TRUE`.

Syntax

```
procedure commit_saga(saga_participant IN VARCHAR2,
 saga_id IN saga_id_t,
 force   IN BOOLEAN DEFAULT FALSE);
```

Parameters

Table 173-4 COMMIT_SAGA Procedure Parameters

Parameter	Description
`saga_participant`	The saga participant (`participant_name`).
`saga_id`	The saga identifier of the type `saga_id_t`.
`force`	The flag to indicate whether the commit saga operation is initiated for a saga branch.

ROLLBACK_SAGA Procedure

This procedure aborts the saga corresponding to the `saga_id` parameter.

The `force` option can be used to forcefully roll back a saga participant transaction for pending or incomplete sagas, For the `ROLLBACK_SAGA()` call on the participant database, `force=TRUE` only rolls back the saga branch operation on the participant database. The default value of `force` is `TRUE`.

Syntax

```
procedure rollback_saga (saga_participant IN VARCHAR2,
 saga_id IN saga_id_t,
 force   IN BOOLEAN DEFAULT FALSE);
```

Parameters

Table 173-5 ROLLBACK_SAGA Procedure Parameters

Parameter	Description
`saga_participant`	The saga participant (`participant_name`).
`saga_id`	The saga identifier of the type `saga_id_t`.
`force`	The flag to indicate whether the rollback saga operation is initiated for a participant transaction. The default value is `TRUE`.

IS_INCOMPLETE Function

This function returns `TRUE` if the saga corresponding to the provided `saga_id` is considered incomplete.

A saga is considered incomplete for the following reasons:

- If the saga exceeds the duration identified by the `max_saga_duration` parameter or the duration established by the `begin_saga()` call.
- If the saga participant fails to finalize the saga (commit or rollback).

Syntax

```
function is_incomplete (saga_id  IN saga_id_t)    returns boolean;
```

Parameters

Table 173-6 IS_COMPLETE Function Parameters

Parameter	Description
`saga_id`	The saga identifier of the type `saga_id_t`.

SET_INCOMPLETE Procedure

This procedure marks the saga corresponding to the provided `saga_id` as incomplete.

This is an administrative interface to mark certain sagas as incomplete such that these sagas can be flagged for manual intervention.

Syntax

```
procedure set_incomplete (saga_id  IN saga_id_t);
```

Parameters

Table 173-7 SET_INCOMPLETE Procedure Parameters

Parameter	Description
`saga_id`	The saga identifier of the type `saga_id_t`.

SEND_REQUEST Procedure

This procedure can be used by the client to enroll a participant in the initiated saga.

This procedure abstracts the creation of the saga message, conversion of JSON payloads, and enqueuing the message to the participant. The application developer can manually perform this step as well.

Syntax

```
procedure send_request(saga_id IN saga_id_t,
 recipient    IN VARCHAR2,
 payload      IN JSON DEFAULT NULL,
 saga_version IN NUMBER DEFAULT 1,
 saga_spare   IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 173-8 SEND_REQUEST Procedure Parameters

Parameter	Description
saga_id	The saga identifier that is derived using the BEGIN_SAGA() function.
recipient	The participant to enroll in this saga.
payload	The JSON payload to call request() of the participant being enrolled.
saga_version	The version of the saga framework (default is 1).
saga_spare	The currently unused field.