



Oracle BPEL Process Manager Demonstration

How to create a time scheduler for a BPEL process using the Oracle Database Job scheduler

by Dr. Constantine Steriadis (constantine.steriadis@oracle.com)

A. Overview

This document is a short guide that demonstrates how to create a scheduling mechanism for Oracle BPEL processes. For example, if you would like to have a process being executed every day sometime between 01:00 and 03:00, then perhaps this guide can be helpful for you.

Prerequisites

This example assumes you are familiar with basic BPEL constructs such as activities partner links etc., that you are familiar with the JDeveloper environment and with Oracle Database for executing a specific script. For this example we assume that the following setup is available:

- Oracle Database 10gR2, with SCOTT/TIGER demo schema
- Oracle BPEL PM 10.1.3.1.0

Implementation Steps

In the next pages we will describe the details of this implementation which is made up of 4 basic steps:

1. Create a table that will act as a "queue" for our timer's events
2. Create a stored procedure that will be executed periodically
3. Register the created stored procedure in the Job scheduler
4. Create a BPEL process that will act as a timer "listener"

An alternative scenario would be to use Oracle Advanced Queuing (AQ) instead of a table, but this will be incorporated in this guide at a later time.

B. Create database table TIMER_TABLE

- Open JDeveloper and create a database connection to Oracle Database named scott. To do that right-click on "Databases" in the Connections tab, select "New Database Connection" and insert the following (adjust the bold to your configuration):

Connection name	: ORCL10g_SCOTT
Username	: scott
Password	: tiger
Hostname / JDBC Port / SID	: e.g. DBhost.mydomain.com / 1521 / ORCL

- Right click on the database connection you have just created, select "SQL Worksheet" and execute the following SQL statements:

<pre>CREATE TABLE "SCOTT"."TIMER_TABLE" ("IDX" NUMBER(10),</pre>

```

"T_CODE" NUMBER(5),
"XDATE" DATE,
"RFLAG" NUMBER(1) DEFAULT 1,
CHECK ("IDX" IS NOT NULL) VALIDATE ,
PRIMARY KEY ("IDX") VALIDATE ) ;

CREATE SEQUENCE "SCOTT"."TIMER_TABLE_SQ" NOCYCLE NOORDER NOCACHE
NOMAXVALUE MINVALUE 1 INCREMENT BY 1 START WITH 1;

CREATE OR REPLACE TRIGGER "SCOTT"."TIMER_TABLE_TRG" BEFORE
INSERT ON "SCOTT"."TIMER_TABLE" REFERENCING OLD AS OLD NEW AS NEW FOR
EACH ROW declare
    mm number;
    dd date;
BEGIN
    select "SCOTT"."TIMER_TABLE_SQ".NEXTVAL into mm from dual;
    select SYSDATE into dd from dual;
    :new.idx := mm;
    :new.xdate := dd;
END;
```

The above statements will create a table named `TIMER_TABLE`, a sequence named `TIMER_TABLE_SQ` to produce unique values to be used as primary keys and a trigger named `TIMER_TABLE_TRG`.

C. Create stored procedure `TIMER_PROC`

- It is generally a good practice to have your procedures inside PL/SQL packages. Therefore we will create a PL/SQL package in `SCOTT` schema with the name `TIMER_PCK` and it will include our stored procedure named `TIMER_PROC`. `TIMER_PROC` will be invoked by the Job scheduler and will perform all the necessary logic, so as to fill `TIMER_TABLE` with a new row, whenever the timer requirements for a new BPEL Process execution is required. In our example these requirements are that the time must be between 01:00 and 03:00 once per day. So as you did before, please right click on the database you have just created, select "SQL Worksheet" and execute the following SQL statements:

```

CREATE OR REPLACE PACKAGE BODY "SCOTT"."TIMER_PCK" AS
    PROCEDURE TIMER_SUBPROC_01
    /* this is a task that will be fired once a day between 0:55 and 03:00
    The exact time of execution depends on when TIMER_PROC will be
    called, which is dependent on how the Database Job
    Scheduler has been set. So the safest thing is to
    set the job scheduler running in short intervals,
    e.g. every 30' and the current procedure will take
    care of having the event fired in the proper timeframe */
    IS
    n1 NUMBER;
    d0 date;
    d1 date;
```

```

d2 date;
d_start date;
d_end date;
BEGIN
    select sysdate into d2 from dual;
    /* calculate timeframe for event */
    d0 := trunc(d2);
    d_start := d0 + 1/24 - 5/(24*60); /* at 00:55 */
    d_end := d0 + 3/24; /* at 03:00 */
    if (d_start < d2)and(d2 < d_end) then
        /* we are in the correct interval */
        /* we must check if event already fired */
        select count(*) into n1 from scott.timer_table where t_code = 1;
        if n1 > 0 then
            /* there is an event, let's check when was it fired */
            select max(xdate) into d1 from scott.timer_table where t_code =
1;
            if d1 < d_start then
                /* must fire the event since this an old one */
                insert into scott.timer_table (t_code) values (1);
                commit;
            end if;
        else
            /* must fire the event since there is no event yet */
            insert into scott.timer_table (t_code) values (1);
            commit;
        end if;
    end if;
END;

/* Universal Timer procedure
this calls all other sub-procedures so that you
can use the same job, for scheduling more than one tasks with
multiple time-profiles. */
PROCEDURE TIMER_PROC IS
BEGIN
    TIMER_SUBPROC_01;
end;

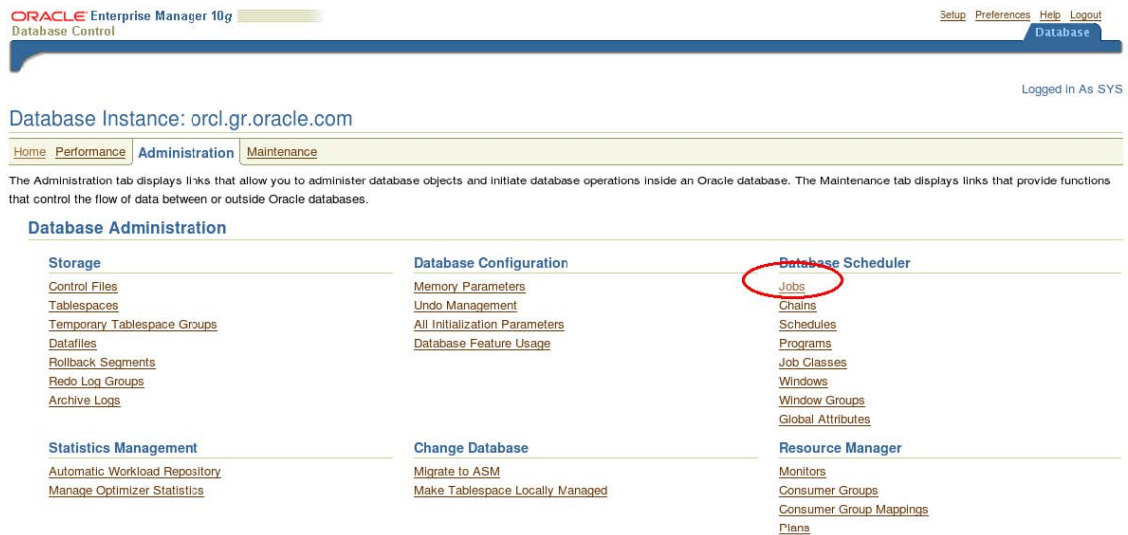
END; /* of package body */

CREATE OR REPLACE PACKAGE "SCOTT"."TIMER_PCK" AS
    PROCEDURE TIMER_PROC;
END; /* of package*/

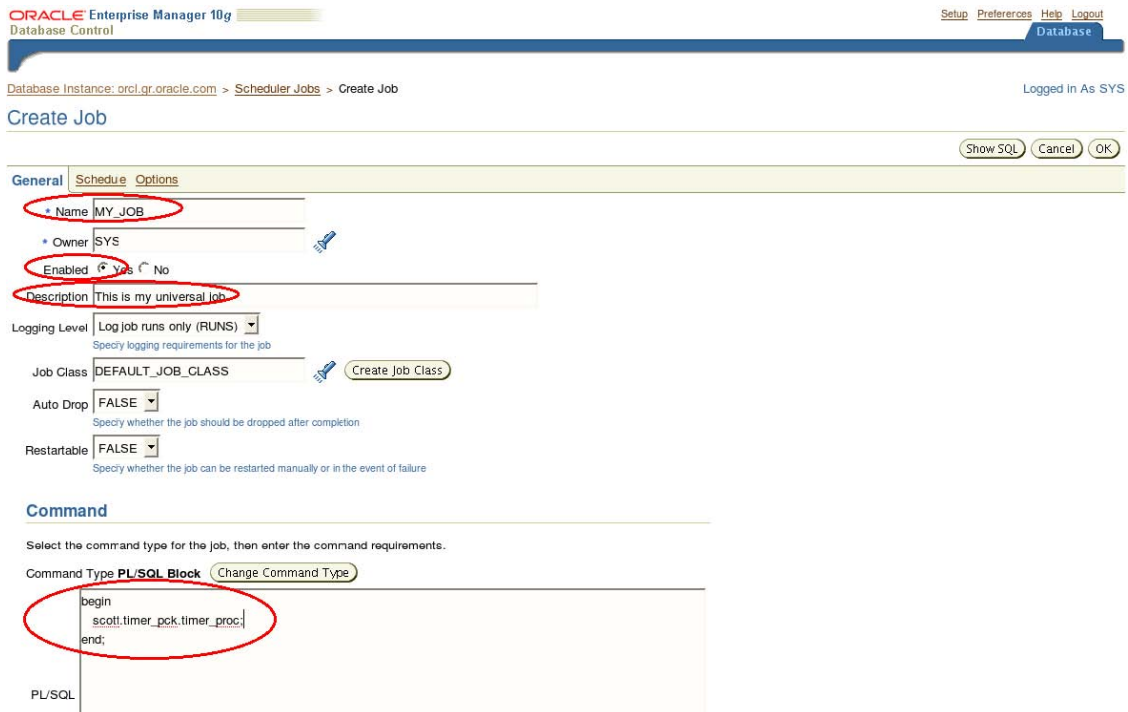
```

D. Register the created stored procedure in the Job scheduler

- Oracle Database enables you to set a “job” to be executed in a repeating fashion. This job can be a PL/SQL code, a stored procedure, etc. So what we need to do is to register the stored procedure created above to the job scheduler, and define the time rules that will be applied in order to have its periodic execution. Job Scheduler has many options, but elaborating more on these is not in the scope of this paper. Now, you will need to login as to the Oracle Database Enterprise Manager through your web-browser (e.g. at DBHost.mydomain.com:1158/em). Please login in as SYS user with SYSDBA privileges.
- Hit the Administrator's tab and select the Jobs link as indicated below:



- Hit the Create button in the page that appears. This will enter you in a page to define the new job. Fill-in the fields of the page with the data indicated below with red:



- Then hit on the Schedule link and fill-in the information as below. When finished press OK.

ORACLE Enterprise Manager 10g
Database Control

Database Instance: orcl.gr.oracle.com > Scheduler Jobs > Create Job

Logged in As SYS

Create Job

Show SQL Cancel OK

General Schedule Options

Schedule Type Standard

Time Zone Africa/Cairo

Repeating

Repeat By Minutes

Interval (Minutes) 30

Available to Start

Immediately

Later

Date Jan 14, 2007
(example: Jan 14, 2007)

Time 4 10 00 AM PM

Not Available After

No End Date

Specified End Date

Date Jan 14, 2007
(example: Jan 14, 2007)


Time 4 20 00 AM PM

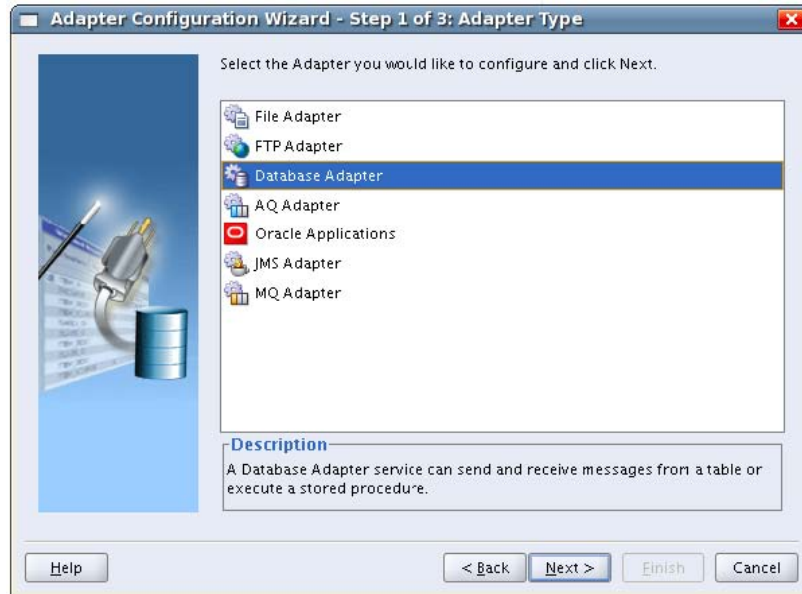
General Schedule Options

E. Create a BPEL process that will act as a timer "listener"

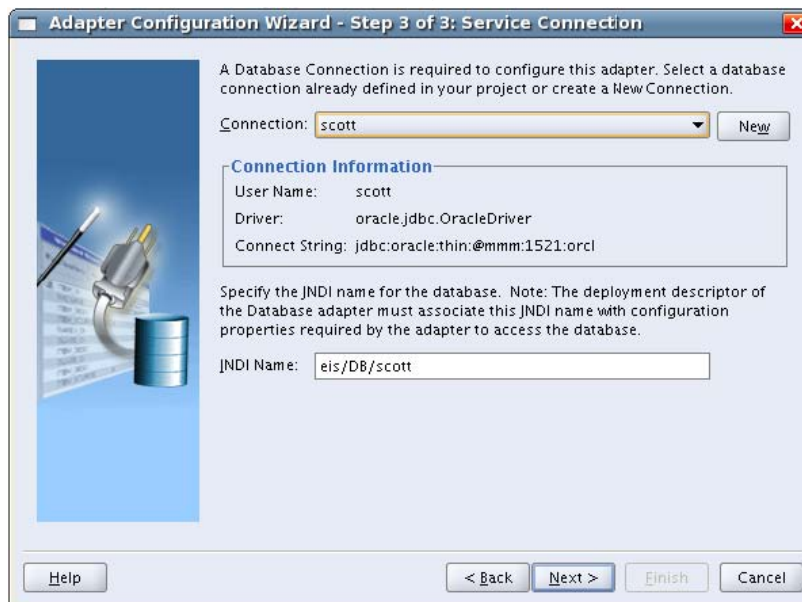
This is the last part of the process. Having created the PL/SQL code that will guarantee to have a new row at specific time intervals in the TIME_TABLE table, what is now necessary is to build-up a BPEL process that will poll the specific table for new rows, flagged with T_Code=1. This is quite easy in JDeveloper 10.1.3.1.0.

Start the JDeveloper 10.1.3.1 by selecting the appropriate icon from the Windows Start Menu as it appears in the image on the right. This is the environment that you will use to design your BPEL process.

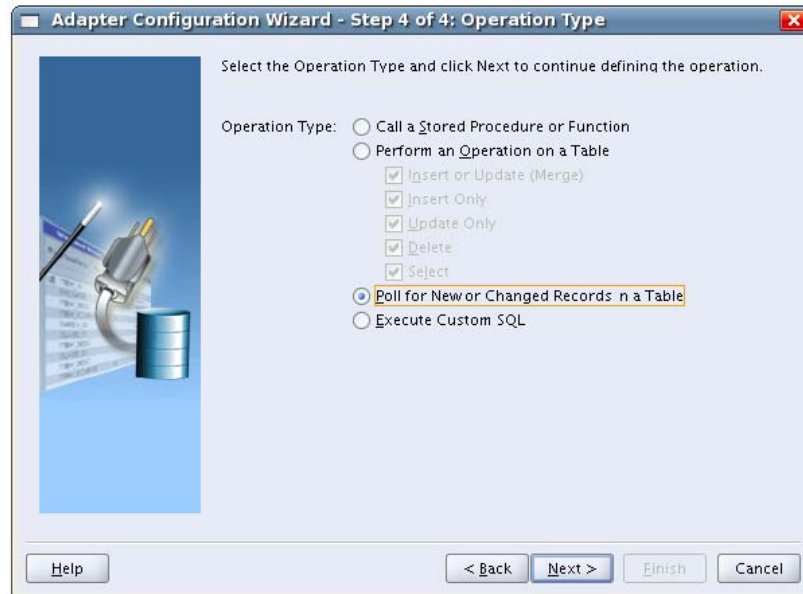
- Create a new Application Workspace in JDeveloper BPEL Designer.
- Create a BPEL project (choosing the "Empty Business Process" template) called BPELTimerProcess. This step creates a directory with BPEL process name in your workspace.
- The inbound DB Adapter component will poll for new rows in the TIMER_TABLE. We will create a partner link to enable the DB Adapter to publish such a new row to the BPEL process.
- Right click on the left swim lane and choose the "Create Partner Link" option. Rename the partner link to "InboundPL".
- We now need to create a WSDL file corresponding to this Partner link. Click on Adapter Service Wizard button  (third button from left) in WSDL setting pane. This will bring up the Adapter services wizard, which allows us to define the DB Adapter service.



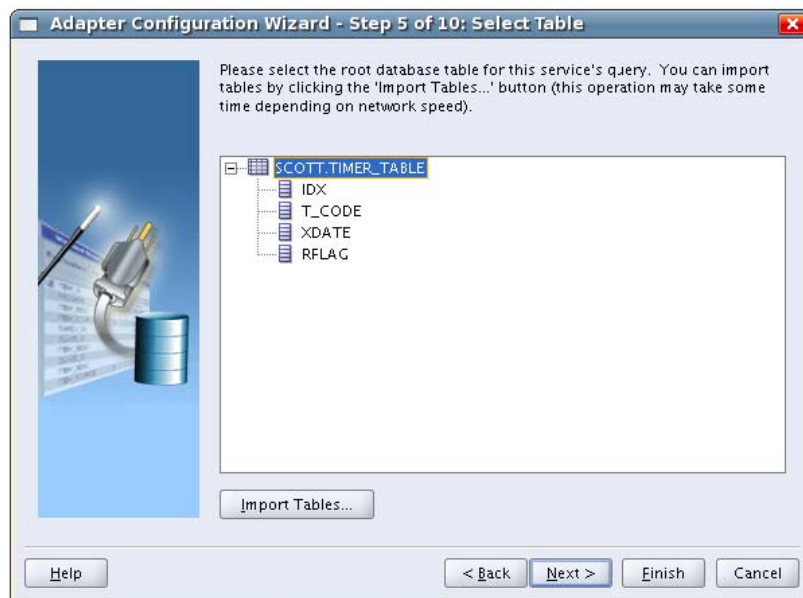
- Click Next, in the Adapter Service Wizard and define the service name as ReadTable. Then click Next and select the database connection you have created before:



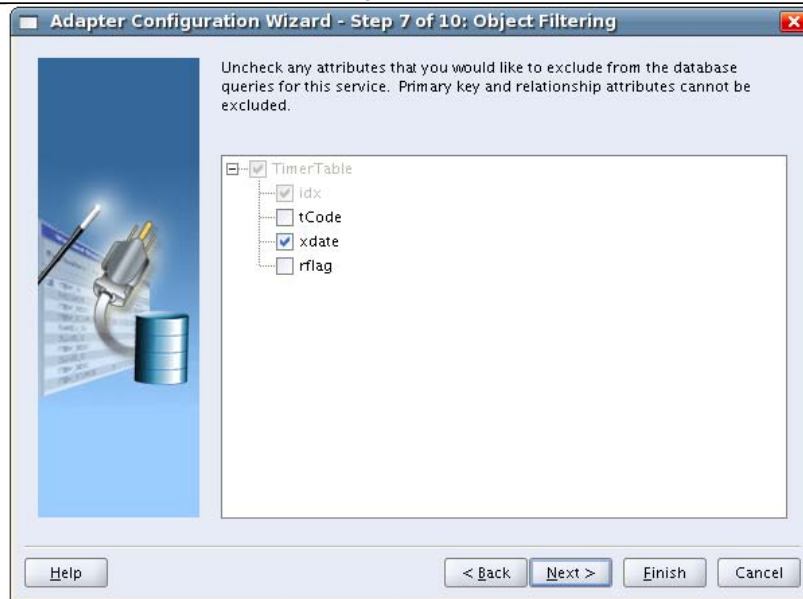
- Click Next in the upper screen and select the option Poll for New or Changed Records in a Table as below. Click Next afterwards. This actually defines the use of the database adapter that you intend to do and besides polling for new records, you can call a stored procedure, make a selection, insert, update, delete or even execute custom SQL query.



- In the next screen you must select the database table that you are interested in. To do this click on the import tables button, press the query button and select the table `TIMER_TABLE` of the `SCOTT` schema. Afterwards press OK:



- Click Next in the relationships definition screen. After that you will be asked to define the table fields to be selected. Depending on the type of data that you will need as an input for your process you can select the fields you are interested in the xDate field, so leave all other unchecked, as appears in the image below and press Next.



- The Database adapter needs to have a way to distinguish which row is new, which is old and which is already read. There are many ways to do this, as appears in the following screen, such as delete read records, logically delete read records by marking them, or using external structures (sequences, files, tables) to mark read records. We will use the logical delete method, which means that a special field of TIMER_TABLE, the rFlag field, will be used for this. So in the current screen select the second option and hit Next.



- In this screen we will define the values that rFlag will take and their meaning. Please note that in the creation script of the table we have defined **1** as default value of rFlag, which means that this should be the unread value. Please fill in the required fields in this screen as below and press Next.

Adapter Configuration Wizard - Step 9 of 11: Logical Delete

Specify the field that should be updated to logically delete the row, and the value to insert in the field to indicate that the row has been read. You can also optionally specify values that indicate if a row is Unread or Reserved.

Logical Delete Field: RFLAG

Read Value: 0

Unread Value: 1

Reserved Value: 2

Help < Back Next > Finish Cancel

- In the next screen you need to define some parameters for the polling behavior, such as what the polling frequency will be, how many rows to be reserved upon polling read, when will a transaction occur, etc. A high polling frequency (e.g. Period of 10 seconds) will generate the load of a query on the TIMER_TABLE every 10 seconds, which may affect the overall database performance. Since our Job Scheduler period is 30 minutes, we cannot expect changes in less than 30 minutes. Given that the job scheduler and the DB adapter are not necessarily synchronized, we can assume that a reasonable polling period would be 15 minutes. So, please enter the values that appear below and press Next

Adapter Configuration Wizard - Step 10 of 11: Polling Options

Specify any additional polling options. To use the recommended, default values, click Next.

Polling Frequency: 15 minutes

Database Rows per XML Document: 1

Database Rows per Transaction: Unlimited 1

Order By: IDX

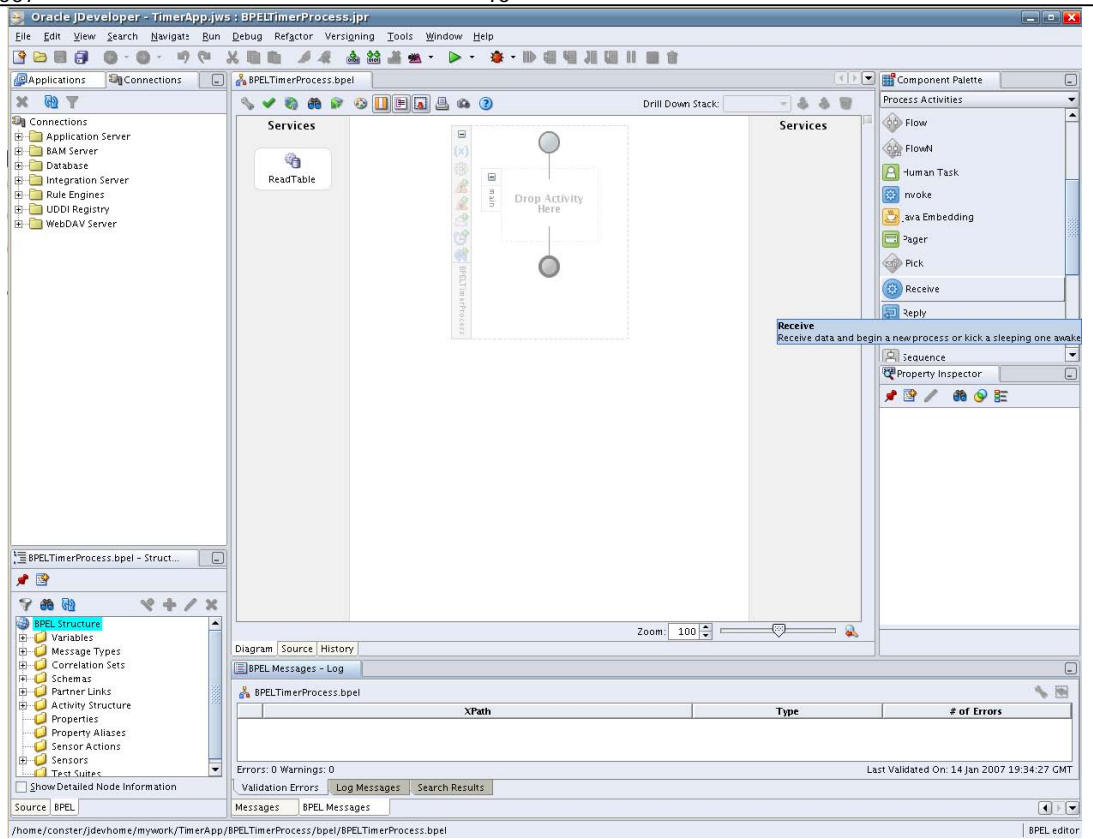
Delay Commit Distributed Polling

Poll for Child Updates Use Batch Destroy

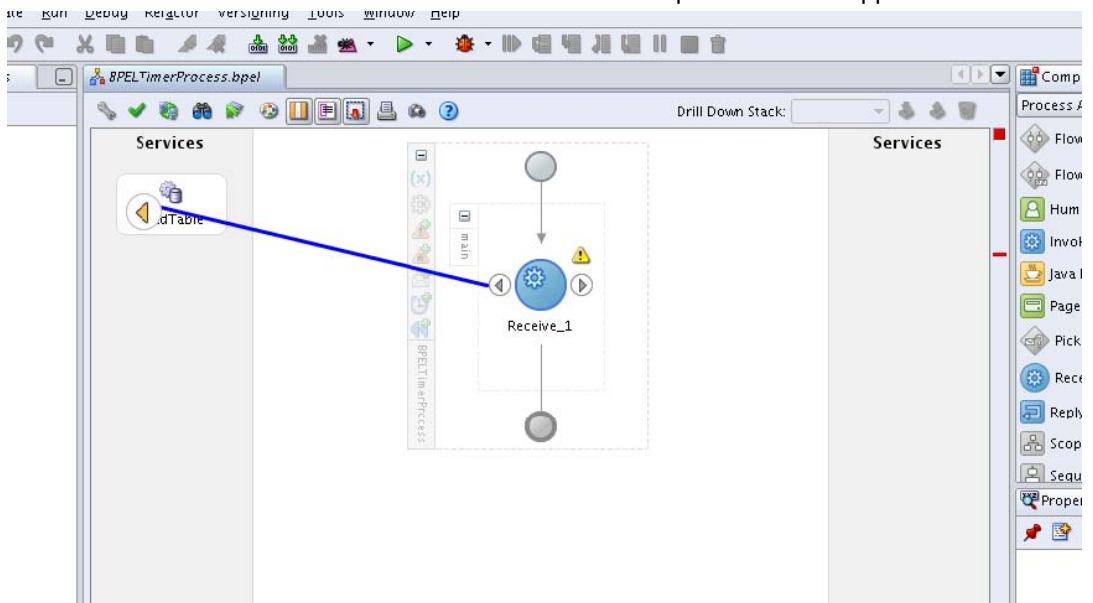
SQL:
SELECT IDX, XDATE FROM TIMER_TABLE WHERE (RFLAG = '2') ORDER BY IDX ASC

Help < Back Next > Finish Cancel

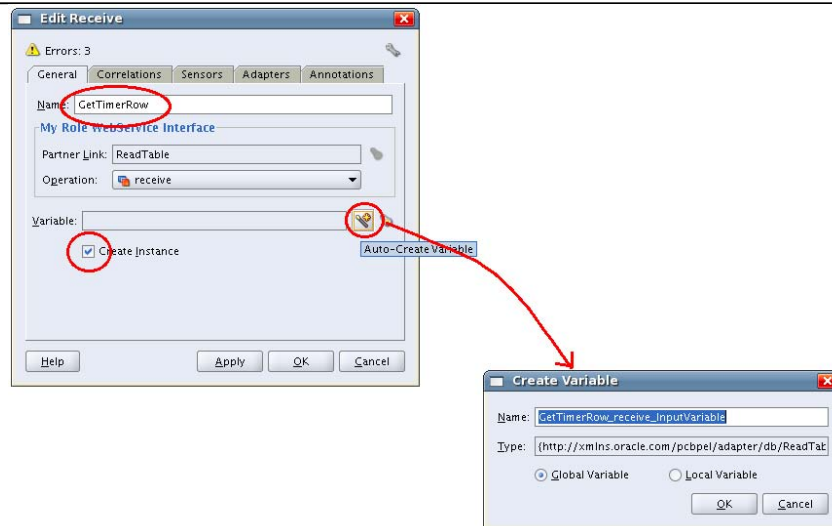
- Press Finish afterwards to complete the database adapter configuration and return to the PartnerLink configuration screen, and then press OK. In JDeveloper you should have the following view:



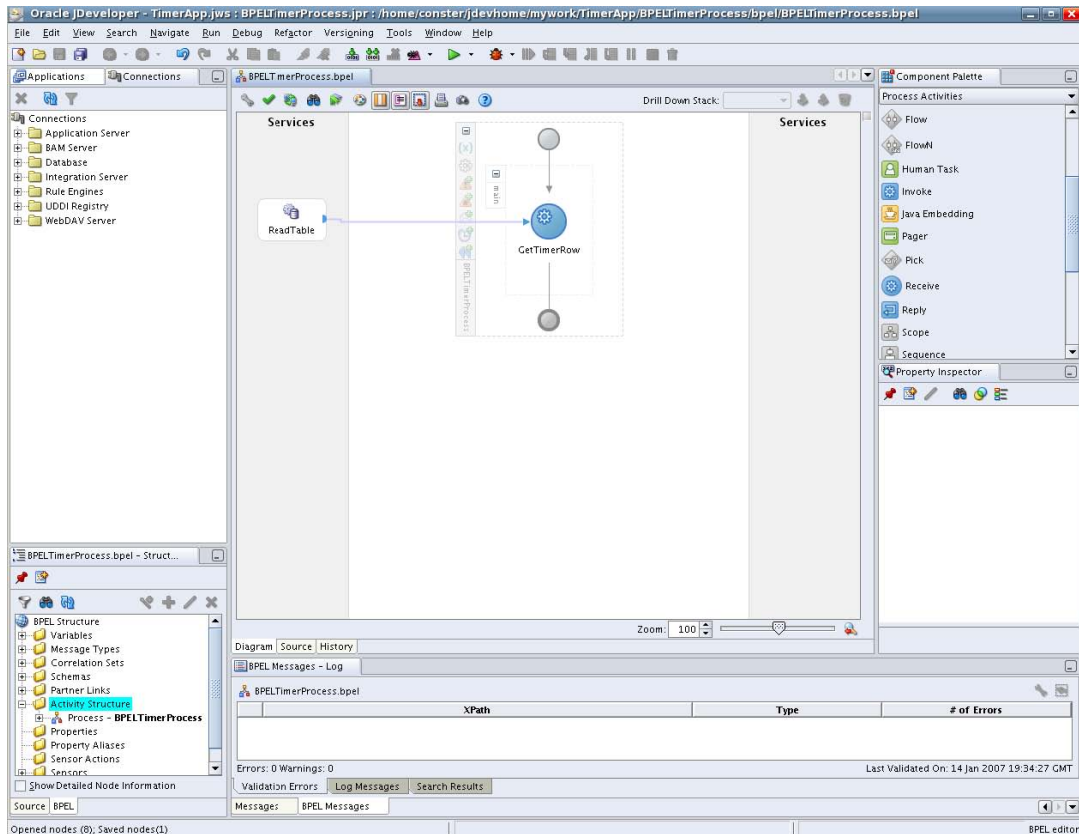
- Drag & drop in the center of the screen the Receive Activity from the upper right panel. After that draw a line with the mouse towards the ReadTable partnerLink as appears below:



- Then the Edit box for the receive activity will appear. Fill in the values as below. The receive activity will need a variable to copy the read values in it. This will be auto-created if you press the magic stick icon on the right and leave all default settings:



- For the final BPEL process you should have the following image:



F. Next steps

So far we have created a BPEL process that will be scheduled to be executed once a day between 01:00 and 03:00. Of course the actual process doesn't do anything more than receive the TIMER_TABLE data, start and finish immediately, which is not very meaningful. In any real case scenario you will need to either design the process that you want inside the BPELTimerProcess, or you can create another process according to your needs and then invoke it from BPELTimerProcess as a partnerLink. This will not be described in this document since it is not in it's scope. Oracle distributes Oracle BPEL PM with a large number of already pre-designed and ready-to-use examples (with their sources) in order to be used in building solutions that fit your needs. These examples cover a wide range of connectivity techniques, or other issues, such as email notifications, Web Services usage, JMS usage, data transformations, etc, and are placed

by default in the folder C:\oracle\BPEL_PMHome\integration\bpelpm\orabpel\samples, during the installation process. If you need more technical information on using Oracle BPEL PM for your solutions you can visit Oracle Technology Network (OTN) at <http://otn.oracle.com/bpel>, relevant public discussion forums at <http://forums.oracle.com/forums> (goto Technologies - > BPEL), or even contact Oracle at <http://www.oracle.com>.

Finally, you can visit Oracle at <http://www.oracle.com/solutions/integration> in order to find more information on Oracle Business Integration Solutions.