

Requirements of Oracle Stored Procedures

In order for Crystal Reports to report off Oracle stored procedures, all of the following requirements must be satisfied:

- To use an ODBC connection to access an Oracle stored procedure, you must create a package that defines the REF CURSOR. This REF CURSOR must be strongly bound to a static pre-defined structure (see Strongly Bound REF CURSORS vs. Weakly Bound REF CURSORS). This package must be created separately and before the creation of the stored procedure.
- The stored procedure must have a parameter that is a REF CURSOR type. Crystal Reports uses this parameter to access and define the result set that the stored procedure returns.
- The REF CURSOR parameter must be defined as IN OUT (read/write mode). After the stored procedure has opened and assigned a query to the REF CURSOR, Crystal Reports will perform a FETCH call for every row from the query's result. This is why the parameter must be defined as IN OUT.
- The parameters can only be defined as IN (input) parameters. Crystal Reports is not designed to work with OUT parameters.
- The REF CURSOR variable must be opened and assigned its query within the procedure.
- The stored procedure can only return one record set. The structure of this record set must not change based on parameters.
- The stored procedure cannot call another stored procedure.
- If you are using the CR ODBC driver, verify that the option **Procedure Return Results** is selected as **On** in the ODBC Driver Configuration setup under the **Advanced** tab.
- If you are using the native Oracle driver and using hard-coded date selection within the stored procedure, the date selection must use either a string representation format of YYYY-DD-MM (where the date field = 2004-01-01) or the **To_Date** formula function with the same format specified (where date field = To_Date(2004-01-01', 'YYYY-MM-DD')).
- Most importantly, the stored procedure must be able to execute successfully in Oracle's SQL *Plus utility.

If all of these requirements have been met, verify that the database driver that you are using works with that version of Oracle.

Creating an Oracle Stored Procedure

Sample A: Use with ODBC and native connections in Crystal Reports

1. Create a table.

```
CREATE TABLE Test_Table
(ID number(5),
Firstname varchar2(30),
Lastname varchar2(30),
Birthday date);
```

2. Insert values into the table.

```

INSERT INTO Test_Table VALUES
(1, 'Christopher', 'Jones', '01-Nov-70');
INSERT INTO Test_Table VALUES
(2, 'Maria', 'Marshall', '02-Jan-77');
INSERT INTO Test_Table VALUES
(3, 'Jonathan', 'Campbell', '09-Aug-75');
INSERT INTO Test_Table VALUES
(4, 'Julie', 'Gagnon', '23-Dec-72');
INSERT INTO Test_Table VALUES
(5, 'Daemon', 'Thompson', '11-Feb-69');

```

3. Create the Package.

```

CREATE OR REPLACE PACKAGE Test_Package
AS TYPE Test_Type IS REF CURSOR RETURN Test_Table%ROWTYPE;
END Test_Package;
/

```

4. Create the stored procedure.

```

CREATE OR REPLACE PROCEDURE Test_Procedure (
Test_Cursor IN OUT Test_Package.Test_Type,
Test_Parameter IN Test_Table.ID%TYPE)
AS
BEGIN
OPEN Test_Cursor FOR
SELECT *
FROM Test_Table
WHERE Test_Table.ID = Test_Parameter;
END Test_Procedure;
/

```

5. Once the stored procedure is successfully created on your Oracle database, execute the stored procedure.

Sample B: Weakly Bound REF CURSOR

Only use this sample with a native connection to Crystal Reports 9 or later.

1. Create a table.

```

CREATE TABLE Test_Table
(ID number(5),
Firstname varchar2(30),

```

**LastName varchar2(30),
Birthday date);**

2. Insert values into the table.

```
INSERT INTO Test_Table VALUES  
(1, 'Christopher', 'Jones', '01-Nov-70');  
INSERT INTO Test_Table VALUES  
(2, 'Maria', 'Marshall', '02-Jan-77');  
INSERT INTO Test_Table VALUES  
(3, 'Jonathan', 'Campbell', '09-Aug-75');  
INSERT INTO Test_Table VALUES  
(4, 'Julie', 'Gagnon', '23-Dec-72');  
INSERT INTO Test_Table VALUES  
(5, 'Daemon', 'Thompson', '11-Feb-69');
```

3. Create the Package.

```
CREATE OR REPLACE PACKAGE Test_Package  
AS TYPE Test_Type IS REF CURSOR;  
END Test_Package;  
/
```

4. Create the stored procedure.

```
CREATE OR REPLACE PROCEDURE Test_Procedure (  
Test_Cursor IN OUT Test_Package.Test_Type,  
Test_Parameter IN Test_Table.ID%TYPE)  
AS  
BEGIN  
OPEN Test_Cursor FOR  
SELECT *  
FROM Test_Table  
WHERE Test_Table.ID = Test_Parameter;  
END Test_Procedure;  
/
```

5. Once the stored procedure is successfully created on your Oracle database, execute the stored procedure.

Sample C: Stored procedure created within a package

Only use this sample with a native connection to Crystal Reports 9 or later.

1. Create a table.

CREATE TABLE Test_Table

```
(ID number(5),  
  
Firstname varchar2(30),  
Lastname varchar2(30),  
Birthday date);
```

2. Insert values into the table.

```
INSERT INTO Test_Table VALUES  
(1, 'Christopher', 'Jones', '01-Nov-70');  
INSERT INTO Test_Table VALUES  
(2, 'Maria', 'Marshall', '02-Jan-77');  
INSERT INTO Test_Table VALUES  
(3, 'Jonathan', 'Campbell', '09-Aug-75');  
INSERT INTO Test_Table VALUES  
(4, 'Julie', 'Gagnon', '23-Dec-72');  
INSERT INTO Test_Table VALUES  
(5, 'Daemon', 'Thompson', '11-Feb-69');
```

3. Create the Package.

```
CREATE OR REPLACE PACKAGE Test_Package  
AS TYPE Test_Type IS REF CURSOR RETURN Test_Table%ROWTYPE;  
PROCEDURE Test_Procedure (  
Test_Cursor IN OUT Test_Type,  
Test_Parameter IN Test_Table.ID%TYPE  
);  
END Test_Package;  
  
/
```

4. Create the Package Body (With The Code For The Stored Procedure).

```
CREATE OR REPLACE PACKAGE BODY Test_Package  
AS  
PROCEDURE Test_Procedure (  
Test_Cursor IN OUT Test_Type,  
Test_Parameter IN Test_Table.ID%TYPE  
) IS  
BEGIN  
OPEN Test_Cursor FOR  
SELECT *  
FROM Test_Table
```

```
WHERE Test_Table.ID = Test_Parameter;  
END Test_Procedure;  
END Test_Package;
```

5. Once the stored procedure is successfully created on your Oracle database, execute the stored procedure.