

Crystal Reports

Oracle Stored Procedures and Crystal Reports

Overview

This article provides information for Oracle stored procedures and Crystal Reports (CR). You will find detailed information on the following:

- CR versions that support Oracle stored procedures.
- Meeting requirements when reporting from Oracle stored procedures in CR.
- Creating Oracle stored procedures.
- Executing Oracle stored procedures in SQL*Plus.

Contents

INTRODUCTION	2
CR VERSIONS THAT SUPPORT ORACLE STORED PROCEDURES	2
<i>CR 9 support</i>	2
<i>CR 8.5 support</i>	2
<i>CR 8 support</i>	3
<i>CR 7 support</i>	3
<i>CR 6 support</i>	4
<i>CR 5 support</i>	4
<i>CR 4 support</i>	5
REQUIREMENTS TO ACCESS ORACLE STORED PROCEDURES FROM CR	5
STRONGLY BOUND REF CURSORS VS WEAKLY BOUND REF CURSORS ..	6
CREATING AN ORACLE STORED PROCEDURE	7
<i>Sample 1 (Use With ODBC and Native Connections in CR)</i>	7
<i>Sample 2: Weakly Bound REF CURSOR (Use With Native Connection in CR 8.5 or Higher Only)</i>	8
<i>Sample 3: Stored Procedure Created Within A Package (Use With Native Connection in CR 9 Only)</i>	9
EXECUTING AN ORACLE STORED PROCEDURE IN SQL*PLUS	11
CREATING A CRYSTAL REPORT OFF AN ORACLE STORED PROCEDURE	11
FINDING MORE INFORMATION	12
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	12

Introduction

Most versions of CR support Oracle stored procedures. This document contains detailed information of which CR versions support Oracle stored procedures.

You will also find information such as meeting requirements when reporting from an Oracle stored procedure in CR, creating Oracle stored procedures, and executing Oracle stored procedures in SQL*Plus.

NOTE	Oracle stored procedures will not function correctly through OLE DB.
-------------	--

CR versions that support Oracle stored procedures

The tables contained in this section list the version of CR that support Oracle stored procedures by native or ODBC connection.

CR 9 support

CR 9 can report off Oracle stored procedures using the following database drivers for the specified version of Oracle:

32-bit only	Oracle 7.x	Oracle 8.05	Oracle 8.1.x	Oracle 9.x
Native Oracle Driver (Crdb_oracle.dll)	No	No	Yes	Yes
CR Oracle 7 ODBC Driver	No	No	No	No
CR Oracle 8 ODBC Driver	No	Yes	Yes	Yes

CR 8.5 support

CR 8.5 can report off Oracle stored procedures using the following database drivers for the specified version of Oracle:

32-bit only	Oracle 7.x	Oracle 8.05	Oracle 8.1.x
Native Oracle Driver (P2sora7.dll)	Yes	Yes	Yes
CR Oracle 7 ODBC Driver	Yes	No	No

CR Oracle 8 ODBC Driver	No	Yes	Yes
--------------------------------	-----------	------------	------------

NOTE	Crystal Reports 8.5 is able to connect to Oracle 9i using the Oracle 9i ODBC driver provided by Oracle. This is the only driver that is supported for a connection between Crystal Reports 8.5 and Oracle 9i.
-------------	---

CR 8 support

CR 8 can report off Oracle stored procedures using the following database drivers for the specified version of Oracle:

32-bit only	Oracle 7.x	Oracle 8.05	Oracle 8.1.x
Native Oracle Driver (P2sora7.dll)	Yes	Yes	Yes (if updated)
CR Oracle 7 ODBC Driver	Yes	No	No
CR Oracle 8 ODBC Driver	No	Yes	Yes (if updated)

Driver updates are available for download at:

<http://www.crystaldecisions.com/downloads>

Search for the following file:

- Cror815.zip

This file contains an updated version of the Oracle 8 database driver (Cror815.dll) for CR, from Merant's 3.6 driver set. This is the first version of the Oracle 8 driver that includes support for Oracle 8i.

- CR8_746.zip
- This updated Oracle native driver (P2sora7.dll) is designed to resolve the empty login error that occurs when attempting to log onto an Oracle server natively with an Oracle 8i client installed.

CR 7 support

CR 7 can report off Oracle stored procedures using the following database drivers for the specified version of Oracle:

16-bit and 32-bit	Oracle 7.x	Oracle 8.05	Oracle 8.1.x
Native Oracle Driver			

(Pdsora7.dll or P2sora7.dll)	Yes	Yes	No
CR Oracle 7 ODBC Driver	Yes	No	No
CR Oracle 8 ODBC Driver	No	Yes	Yes (if updated)

Driver updates are available for download at:

<http://www.crystaldecisions.com/downloads>

Search for the following file:

- Cror815.zip

This file contains an updated version of the Oracle 8 database driver (Cror815.dll) for CR, from Merant's 3.6 driver set. This is the first version of the Oracle 8 driver that includes support for Oracle 8i.

CR 6 support

CR 6 can report off Oracle stored procedures using the following database drivers for the specified version of Oracle:

16-bit and 32-bit	Oracle 7.x	Oracle 8.05	Oracle 8.1.x
Native Oracle Driver (Pdsora7.dll or P2sora7.dll)	Yes	Yes	No
CR Oracle 7 ODBC Driver	No	No	No

CR 5 support

CR 5 can report from Oracle stored procedures using the following database drivers for the specified version of Oracle:

16-bit and 32-bit	Oracle 7.x	Oracle 8.05	Oracle 8.1.x
Native Oracle Driver (Pdsora7.dll or P2sora7.dll)	Yes	Yes	No
CR Oracle ODBC Driver	No	No	No

Driver updates are available for download at:

<http://www.crystaldecisions.com/downloads>

Search for the following file:

- Ociw16.zip

This .zip file should only be used for CR version 5.0.x.108, 16-bit user who are using Oracle 7.1 and Oracle 7.2 and are looking to use our 16-bit Native Oracle driver.

- Ora16.zip

These files enable you to install the CR 16-bit ODBC drivers for connecting to Oracle 6. These files only apply when using CR version 5.

- Ora716.zip

These files enable you to install the CR 16-bit ODBC drivers for connecting to Oracle 7. These files only apply when using CR version 5.

- Ora732.zip

These files enable you to install the CR 32-bit ODBC drivers for connecting to Oracle 7. These files only apply when using CR version 5.

CR 4 support

CR 4 does not support Oracle stored procedures. Oracle stored procedures return data through output parameters and CR 4 cannot interpret output parameters.

Requirements to access Oracle stored procedures from CR

The following requirements must be met in order for CR to access an Oracle stored procedure:

1. 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.

NOTE	<p>Crystal Reports 9 native connections will support Oracle stored procedures created within packages as well as Oracle stored procedures referencing weakly bound REF CURSORS.</p> <p>Crystal Reports 8.5 native connections will support Oracle stored procedures referencing weakly bound REF CURSORS.</p>
-------------	---

2. The procedure must have a parameter that is a REF CURSOR type. This is because CR uses this parameter to access and define the result set that the stored procedure returns.

3. The REF CURSOR parameter must be defined as IN OUT (read/write mode). After the procedure has opened and assigned a query to the REF CURSOR, CR will perform a FETCH call for every row from the query's result. This is why the parameter must be defined as IN OUT.
4. Parameters can only be input (IN) parameters. CR is not designed to work with OUT parameters.
5. The REF CURSOR variable must be opened and assigned its query within the procedure.
6. The stored procedure can only return one record set. The structure of this record set must not change, based on parameters.
7. The stored procedure cannot call another stored procedure.
8. If using an ODBC driver, it must be the CR Oracle ODBC driver (installed by CR). Other Oracle ODBC drivers (installed by Microsoft or Oracle) may not function correctly.
9. If you are using the CR ODBC driver, you must ensure that in the ODBC Driver Configuration setup, under the Advanced Tab, the option 'Procedure Return Results' is checked ON.
10. If you are using the native Oracle driver and using hard-coded date selection within the procedure, the date selection must use either a string representation format of 'YYYY-DD-MM' (i.e. WHERE DATEFIELD = '1999-01-01') or the TO_DATE function with the same format specified (i.e. WHERE DATEFIELD = TO_DATE ('1999-01-01','YYYY-MM-DD')). For more information, refer to kbase article C2008023.
11. Most importantly, this stored procedure must execute successfully in Oracle's SQL*Plus utility.

If all of these conditions are met, you must next ensure you are using the appropriate database driver. Please refer to the sections in this white paper for a list of acceptable database drivers.

Strongly Bound REF CURSORS vs Weakly Bound REF CURSORS

When declaring the REF CURSOR in an Oracle package, the REF CURSOR may be strongly bound or it may be weakly bound.

A strongly bound REF CURSOR is bound with the same structure that is used by the outputted SELECT statement in the Oracle stored procedure.

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

The code in green displays the REF CURSOR strongly bound. In this example, the REF CURSOR is bound with the row structure of the Test_Table table.

A weakly bound REF CURSOR is not assigned a structure in the Oracle package.

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

Although the REF CURSOR is not assigned a structure in the Oracle package, the Oracle stored procedure itself must return a record set with a structure that does not change based on parameters.

NOTE

Crystal Reports 8.5 will support Oracle stored procedures that use a weakly bound REF CURSOR. This is only the case for a native connection in Crystal Reports 8.5.

Creating an Oracle stored procedure

The following is a list of steps required to create sample Oracle stored procedures that will return a result set in CR. These scripts should be executed in the native Oracle connection utility, SQL*Plus.

NOTE

These stored procedures are only samples. Please refer to your Oracle documentation for more information on creating stored procedures.

Sample 1 (Use With ODBC and Native Connections in CR)

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 2: Weakly Bound REF CURSOR (Use With Native Connection in CR 8.5 or Higher Only)

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 3: Stored Procedure Created Within A Package (Use With Native Connection in CR 9 Only)

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.

Executing an Oracle stored procedure in SQL*Plus

The following script will execute the sample stored procedure created in this white paper. This script should be run in Oracle's native connection utility, SQL*Plus. Remember to change the <parameter value> to a valid parameter value (between 1 and 5).

```
SET SERVEROUTPUT ON  
  
declare  
    test_cursor test_package.test_type;  
    resultset test_cursor%rowtype;  
begin  
    test_procedure(test_cursor, <parameter value>);  
    if not test_cursor%isopen then  
        dbms_output.put_line('the cursor is not open');  
    else  
        dbms_output.put_line('the cursor is open');  
    end if;  
    fetch test_cursor into resultset;  
    while test_cursor%found loop  
        dbms_output.put_line(resultset.ID);  
        dbms_output.put_line(resultset.Firstname);  
        dbms_output.put_line(resultset.Lastname);  
        dbms_output.put_line(resultset.Birthday);  
        fetch test_cursor into resultset;  
    end loop;  
end;  
/
```

NOTE

If the script is executing correctly but you do not see any results displayed to the screen, enter SET SERVEROUTPUT ON at the SQL> prompt to display results to the screen.

Creating a Crystal Report off an Oracle Stored Procedure

1. In Crystal Reports, click the **File** menu, and then click **New**.

2. Click **As a Blank Report**, and then click **OK**.
3. In the **Database Expert** dialog box, find your data source. Double-click on the data source and you will see a **Stored Procedures** node. Double-click on that and you will see the tables in the stored procedure.
4. Move the table(s) you want to report off into the **Selected Tables** box. If the stored procedure has a parameter you will be prompted to enter a value at this point.
5. Click **OK** to close the **Enter Parameter Values** dialog box, and then click **OK** to close the **Database Expert**.

You can now design your report.

NOTE	If you do not see the Stored Procedures node in the Database Expert do the following. On the File menu, click Options . On the Database tab, select the Stored Procedures check box.
-------------	--

Finding More Information

For more information regarding Oracle stored procedures, search for the kbase article c2001789 at <http://support.crystaldecisions.net/kbase>.

For more information regarding the creation of Oracle Stored procedures please refer to www.oracle.com.

Contacting Crystal Decisions for Technical Support

Along with this document, and the *Crystal Reports User's Guide*, we recommend that you visit our Technical Support web site for further resources and sample files. For further assistance with Oracle stored procedures, visit us at the websites below.

Technical Support web site:

<http://support.crystaldecisions.com/homepage/>

Answers By Email Support:

<http://support.crystaldecisions.com/support/answers.asp>

Phone Support:

Tel: (604) 669-8379