RADventure Oracle Dictionary Tools Programmers Documentation

Table of contents

1 Welcome to the RADventure Oracle Dictionary Tools

After completing a Clarion dictionary for an Oracle database, the first thing that comes to mind is to create the Database Schema. In a Database Schema, all user's objects are collected. Objects are normally stored in tablespaces. The RADventure Oracle Dictionary Tools can create a number of objects automatically for you. It also contains a number of tools that are useful during the development cycle of your database.
The RADventure Oracle Dictionary tools are implemented as a Utility template. This template works with the legacy template chain as well as with the ABC template chain. It is compatible with version C5 and C5.5(g). The version is tested from Oracle version 7.3 and up. Therefore, it is compatible with version 8i and 9i.

1.1 Oracletools Features

Oracletools perform the following tasks.

1) Generate a table create script including references to the primary key and tablespaces for the table and the primary key
(CREATE TABLE)
2) Generate an index creation script
CREATE(INDEX)
3) Generate a Comments col and comment table scripts; this adds your dictionary table and field description to the Oracle data dictionary
(COMMENT ON COLUMN)
4) Generate a table synonym script
(CREATE SYNONYM)
5) Generate a trigger skeleton for your triggers
6) Generate primary and foreign key constraints including CASCADE DELETES
7) Generate a dictionary compare program that compares the dictionary against your Oracle database
8) Generate a drop table script
9) Generate a conversion program from Oracle lite to Oracle

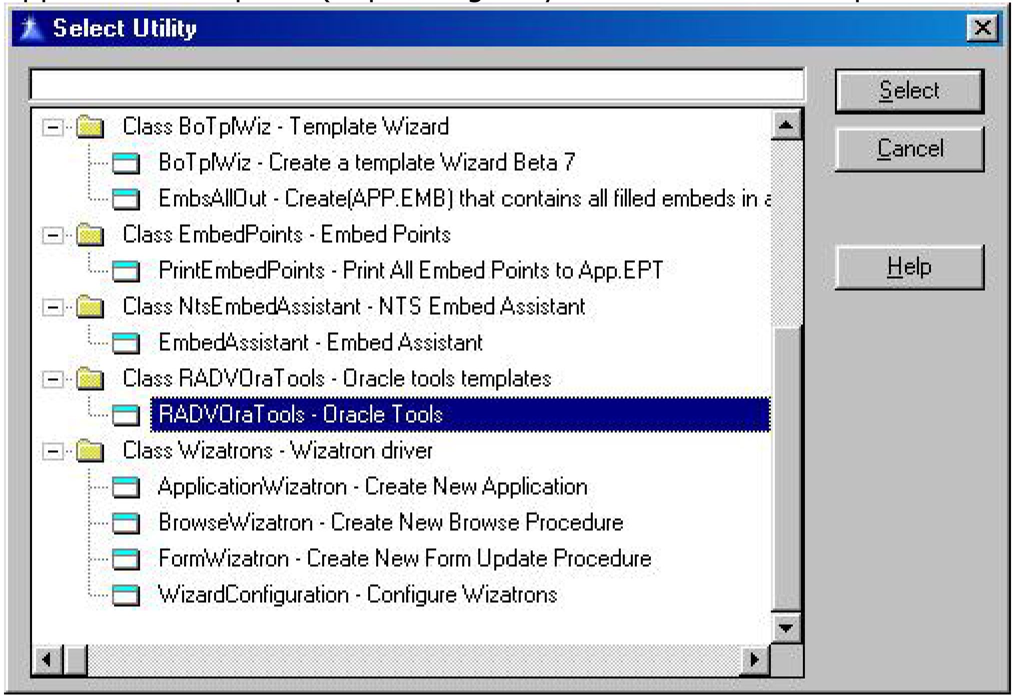
1.2 Installation

To install the Oracletools simply run the setup program. The template is added to the 3rdParty\template directory in accordance with the 3pa directions.

1.3 Starting the Oracletools

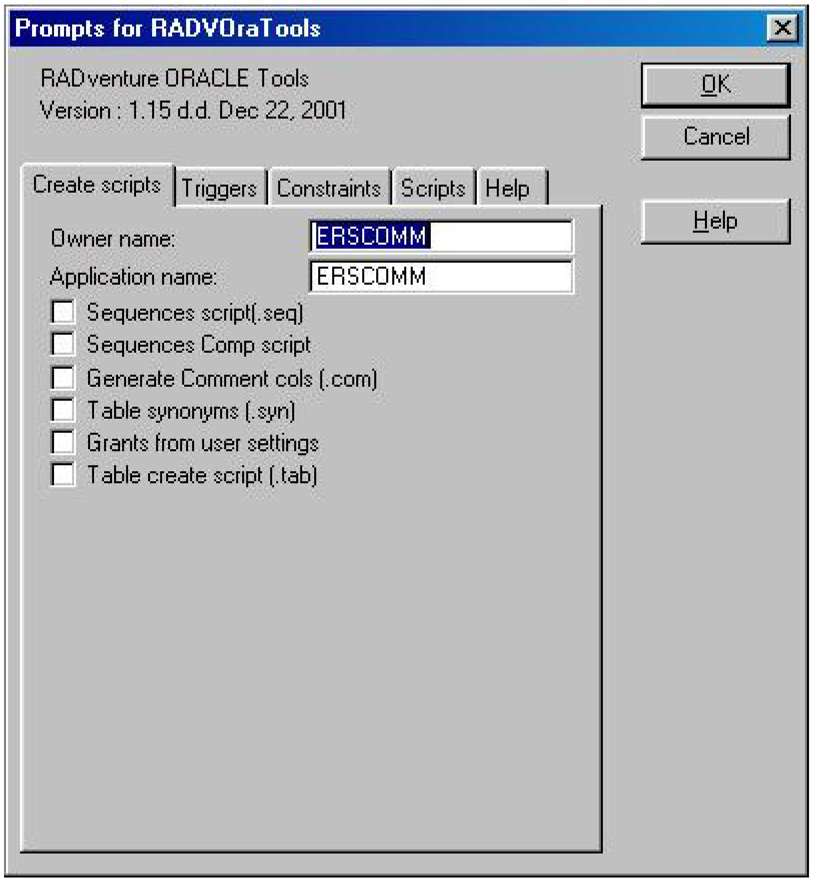
To start the Oracletools do the following.

1) Open any valid Clarion application that contains a dictionary
   a) Open the Clarion IDE
   b) Open your application
2) Click on the menu '**Application**' then on '**Template Utility**'
3) From the available utility templates select :
   RADvoratools - Oracletools
   The following screen representation is a sample of the screen that appears at this point (depending on your own loaded template utilities)



Class BoTplWiz - Template Wizard
BoTplWiz - Create a template Wizard Beta 7
EmbsAllOut - Create(APP.EMB) that contains all filled embeds in a
Class EmbedPoints - Embed Points
PrintEmbedPoints - Print All Embed Points to App.EPT
Class NtsEmbedAssistant - NTS Embed Assistant
EmbedAssistant - Embed Assistant
Class RADVOraTools - Oracle tools templates
RADVOraTools - Oracle Tools
Class Wizatrons - Wizatron driver
ApplicationWizatron - Create New Application
BrowseWizatron - Create New Browse Procedure
FormWizatron - Create New Form Update Procedure
WizardConfiguration - Configure Wizatrons

4 This window should appear.



Prompts for RADVOraTools

RADventure ORACLE Tools
Version : 1.15 d.d. Dec 22, 2001

Owner name: ERSCOMM
Application name: ERSCOMM

- [ ] Sequences script(.seq)
- [ ] Sequences Comp script
- [ ] Generate Comment cols (.com)
- [ ] Table synonyms (.syn)
- [ ] Grants from user settings
- [ ] Table create script (.tab)

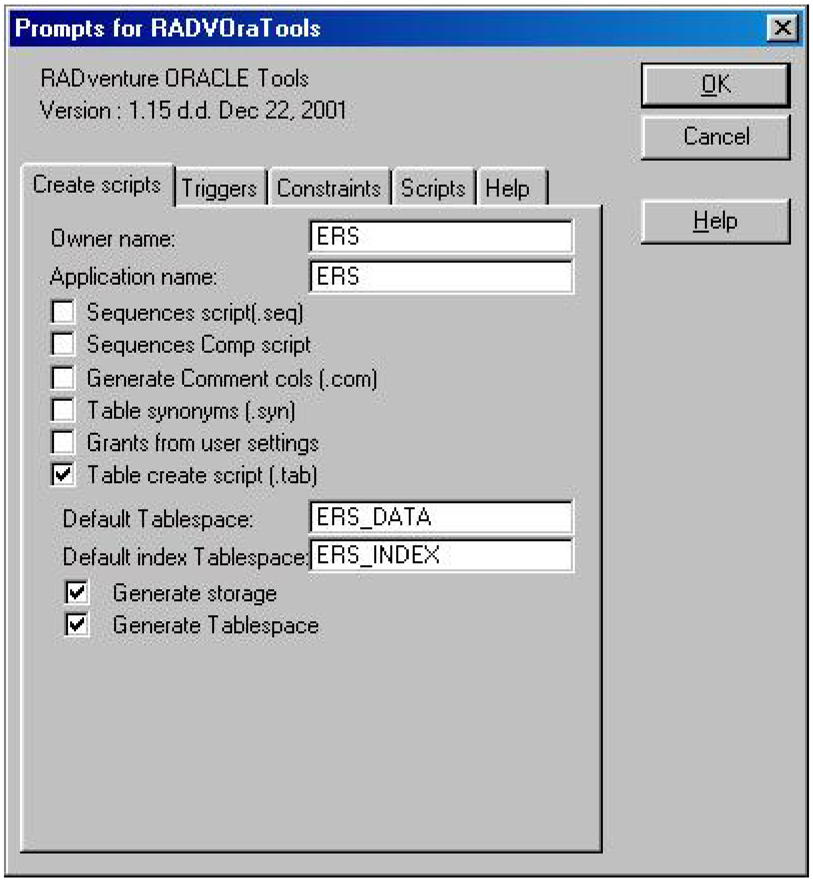
There are a number of options you can set on each window, but they are mostly used for special cases. The name of the owner is used for the scripts at certain points. The name of the application is used, and a file prefix for the scripts is generated. The following script names are used:

For the Sequence script	<appname>.seq
For the table create script	<appname>.tab
For the comment script	<appname>.com
For the sequence compare script	Compseq.sql
For the primary key script	<appname>.pk
For the foreign key script	<appname>.fk
For the index script	<appname>.ind
Drop table script	droptab.sql
Dictionary compare program	Compdict.prj and Compdict.clw

2 Creating different scripts

2.1 Generating a Table Creation Script

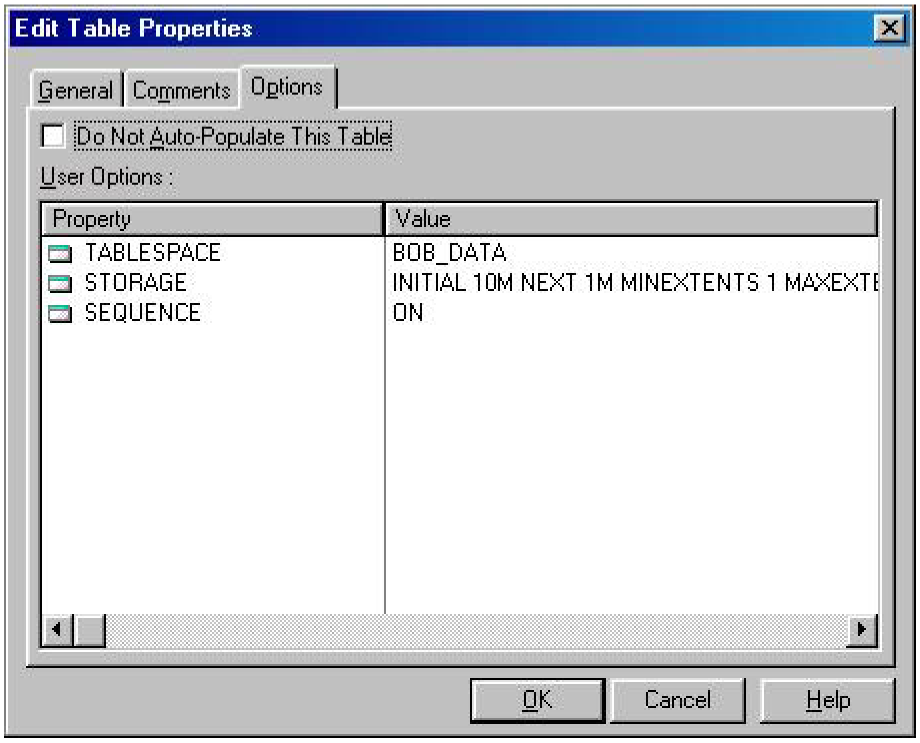
You can generate a table creation script with the option **Table Create script** on tab 1 (**Create scripts**).
When you click this option extra tablespace and storage information is requested.



RADventure ORACLE Tools
Version : 1.15 d.d. Dec 22, 2001

Owner name: ERS
Application name: ERS
☐ Sequences script(.seq)
☐ Sequences Comp script
☐ Generate Comment cols (.com)
☐ Table synonyms (.syn)
☐ Grants from user settings
☑ Table create script (.tab)
Default Tablespace: ERS_DATA
Default index Tablespace: ERS_INDEX
☑ Generate storage
☑ Generate Tablespace

You can indicate a default tablespace for the tables, and you can indicate a separate index tablespace for the primary key constraint. You can change the tablespace per table in the user file options of the dictionary. An example appears in the following screen (take from clarion datadictionary):

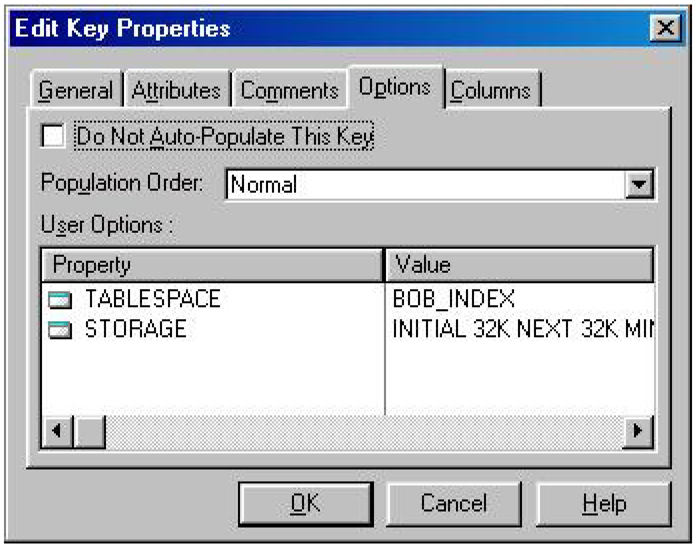


☐ Do Not Auto-Populate This Table

User Options :

Property	Value
TABLESPACE	BOB_DATA
STORAGE	INITIAL 10M NEXT 1M MINEXTENTS 1 MAXEXTE
SEQUENCE	ON

The tablespace for this table is generated as BOB_DATA.
The default storage clause is overwritten with the one as cited in the user option STORAGE.
You can skip generating the table by putting in a user option called CREATETAB. Set as value NO.
You can do the same for the primary key. Go to the KEY user options.



☐ Do Not Auto-Populate This Key

Population Order: Normal

User Options :

Property	Value
TABLESPACE	BOB_INDEX
STORAGE	INITIAL 32K NEXT 32K MI

The Primary key index is generated in table space BOB_INDEX, with a different storage clause.
If you do not indicate any storage clause, the default storage is generated. This is:
STORAGE(INITIAL 32K NEXT 32K MINEXTENTS 1 MAXEXTENTS UNLIMITED PCTINCREASE 0)

2.2 Creating a Sequence Script

When you want to generate autonumbering keys numbered by Oracle you will need a sequence. These sequences can be generated by checking **'Sequence script'**.
This generates a script with a sequence name equal to the file prefix followed by '**_seq**'.
For example:
CREATE SEQUENCE bob.CLI_SEQ START WITH 1 INCREMENT by 1;
Bob is the owner. The file prefix is SEQ.
Sequences are only generated for files with the user option SEQUENCE=ON (see tab in creating table scripts).

2.3 Creating a Sequence Compare Script

During development you will want to see if the maximum values in your autonumbering keys are still in line with the Oracle sequences. To do this create the Compare sequences script.
You can create this script by checking: '**Sequences Comp. Script.'** This generates a compseq.sql which you can execute in SQL plus. It generates two statements:

SELECT MAX(Client_bezwaar_id) from bob.CLIENT_BEZWAAR;
SELECT bob.CBE_SEQ.CURRVAL FROM DUAL;

2.4 Creating a Table Comment Script

Oracle allows you to put column and table comments in the Oracle Data dictionary. At RADventure we use these descriptions for our QBE tool. It shows the user a description of the fields during a QBE session, instead of just the column name.
You can create a comment script for your dictionary by checking: '**Generating comment columns (.com)**'
The resulting file (<appname.com>) contains statements such as the following for your complete dictionary:

COMMENT ON TABLE MEMO_CATEGORIE IS 'Comment categories for your client memos, also for complaints.';
COMMENT ON COLUMN bob.MEMO_CATEGORIE.C_MEMO_CATEGORIE IS 'Code';
COMMENT ON COLUMN bob.MEMO_CATEGORIE.OMS_MEMO_CATEGORIE IS 'Description';

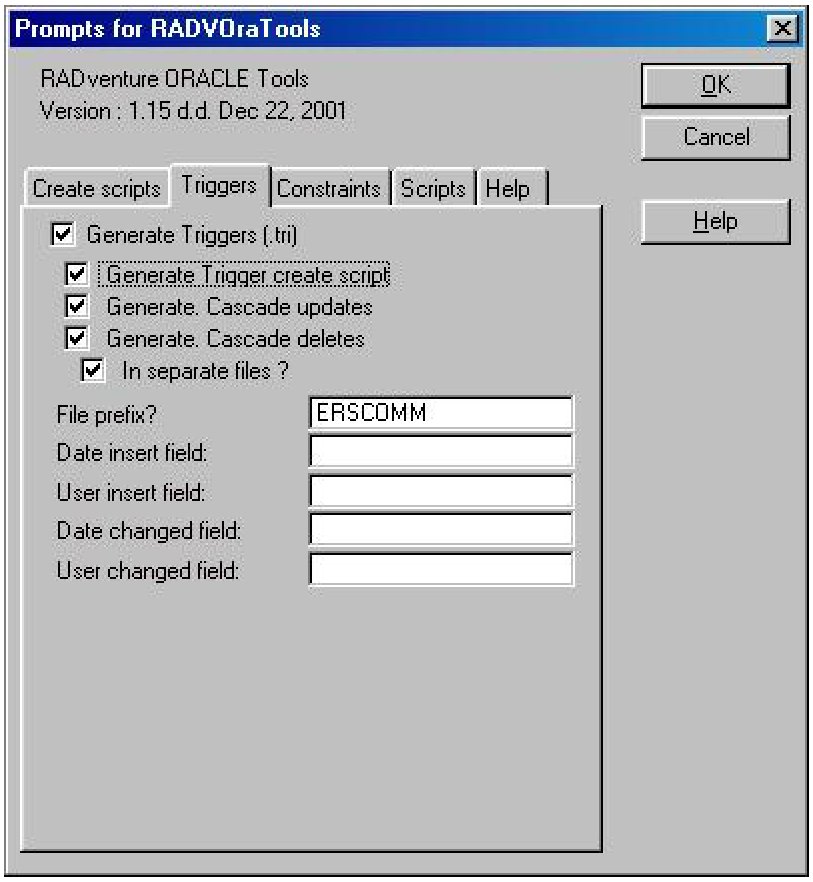
2.5 Creating Public Synonyms for your Tables

You can generate public synonyms for all your tables by checking **'Table synonyms'**. This generates a script <appname>.syn with contents such as the following:

CREATE PUBLIC SYNONYM MEMO_CATEGORIE for bob.MEMO_CATEGORIE; -- Create the synonym for table
CREATE PUBLIC SYNONYM ROL for bob.ROL; -- Create the synonym for table

2.6 Creating Trigger Skeleton Scripts

You can generate trigger skeleton scripts for all your tables. If you do not use declarative foreign key constraints, you can use these scripts to generate cascade updates and cascade deletes.
Select the tab '**Triggers**' . You will see the following window:



RADventure ORACLE Tools
Version : 1.15 d.d. Dec 22, 2001

[✓] Generate Triggers (.tri)
[✓] Generate Trigger create script
[✓] Generate. Cascade updates
[✓] Generate. Cascade deletes
[✓] In separate files ?

File prefix?	ERSCOMM
Date insert field:	
User insert field:	
Date changed field:	
User changed field:	

This template part generates all triggers in one file <appfile>.tri unless you check the option '**In separate files**' . In this case the files are generated in files named
<fileprefix>_<tablename>.tri.

A trigger generated with cascade constraints appears as follows:

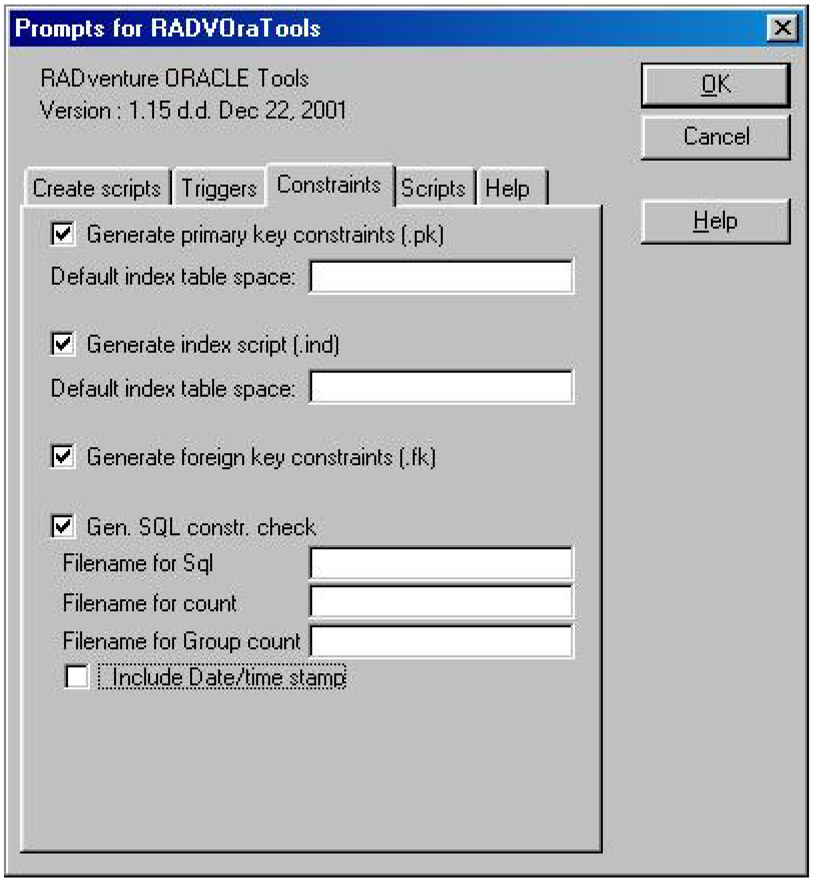
Create or replace trigger IU_CLI before insert or update or delete on CLIENT
for each row
begin

```
if inserting then
   select CLI_seq.nextval into :new.ON from dual;
end if;
if updating then
   If :new.CLIENT_ID <> :old.CLIENT_ID then
      update CLIENT_PROCESINFO set CLIENT_ID=:new.CLIENT_ID where
CLIENT_ID=:old.CLIENT_ID ;
      update CORRESPONDENTIE_CLIENT set CLIENT_ID=:new.CLIENT_ID
where CLIENT_ID=:old.CLIENT_ID ;
      update CLIENT_MEMO set CLIENT_ID=:new.CLIENT_ID where
CLIENT_ID=:old.CLIENT_ID ;
      update CLIENT_STATUS set CLIENT_ID=:new.CLIENT_ID where
CLIENT_ID=:old.CLIENT_ID ;
   end if;
end if;
if deleting then
   delete from CLIENT_PROCESINFO where CLIENT_ID=:old.CLIENT_ID ;
   Alg.Testfordelete('FOLLOW_UP','CLIENT_ID=' || :old.CLIENT_ID);
   delete from CLIENT_BEZWAAR where CLIENT_ID=:old.CLIENT_ID ;
   delete from CORRESPONDENTIE_CLIENT where CLIENT_ID=:old.CLIENT_ID ;
   delete from ONDERZOEK where CLIENT_ID=:old.CLIENT_ID ;
   delete from ARCHIEF where CLIENT_ID=:old.CLIENT_ID ;
   delete from CLIENT_MEMO where CLIENT_ID=:old.CLIENT_ID ;
   delete from CLIENT_STATUS where CLIENT_ID=:old.CLIENT_ID ;
   delete from UITNODIGING where CLIENT_ID=:old.CLIENT_ID ;
end if;
end;
```

The generation process checks if in the relation between files CASCADE SERVER is added for the update, and the delete parts of the relation.

2.7 Creating a Primary/Foreign Key Constraints Script

In the tab '**Constraints**' you can generate the primary and foreign key constraints. The primary key constraint script is generated in <appname>.pk. The '**constraints'** tab shows the different constraint options.



RADventure ORACLE Tools
Version : 1.15 d.d. Dec 22, 2001

- [✓] Generate primary key constraints (.pk)
- Default index table space:
- [✓] Generate index script (.ind)
- Default index table space:
- [✓] Generate foreign key constraints (.fk)
- [✓] Gen. SQL constr. check
- Filename for Sql
- Filename for count
- Filename for Group count
- [ ] Include Date/time stamp

When you generate a primary key constraints file, the file contains a script such as the following:

ALTER TABLE MEMO_CATEGORIE
ADD (CONSTRAINT MCA$P_KEY PRIMARY KEY
(C_MEMO_CATEGORIE)
USING INDEX TABLESPACE BOB_INDEX
STORAGE (INITIAL 32K NEXT 32K MAXEXTENTS UNLIMITED PCTINCREASE 0));

The storage can be changed at your key user options, with the storage clause added as an option.

When you generate a foreign key constraints file, the file <appname>.fk contains statements such as the following:

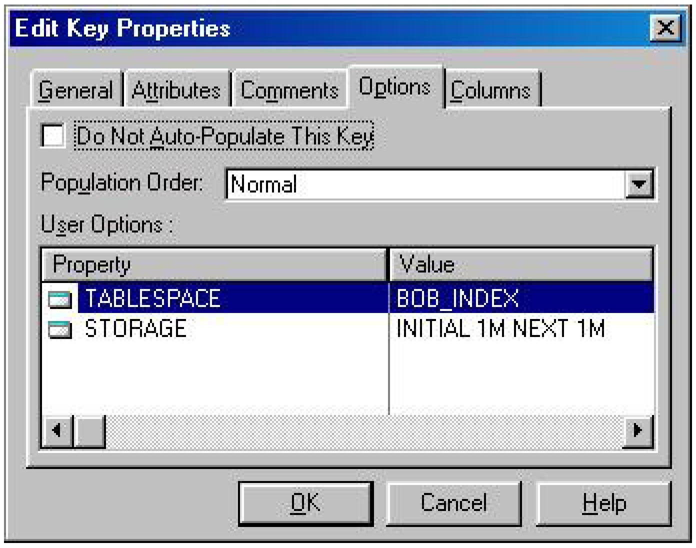
*ALTER TABLE EIND_DIAGNOSE
ADD (CONSTRAINT EID$FK_EID_SST$SST$P_KEY_SST FOREIGN KEY (C_SRT_STATUS)
REFERENCES SOORT_STATUS (C_SRT_STATUS));
ALTER TABLE ARCHIEF
ADD (CONSTRAINT ARC$FK_ARC_OND$OND$P_KEY_OND FOREIGN KEY (ONDERZOEK_ID)
REFERENCES ONDERZOEK (ONDERZOEK_ID)
ON DELETE CASCADE);*

2.8 Creating an Index Script

You can create an index script with the **Gen. index script (.ind)** option on the **constraints** tab. The script contains statements such as the following:

CREATE INDEX IND$EID_FK_EID_SST ON EIND_DIAGNOSE (C_SRT_STATUS ASC) TABLESPACE BOB_INDEX;

A storage clause is added when you add a user option to the key called STORAGE. You can also add a tablespace reference as an option TABLESPACE.



Do Not Auto-Populate This Key [ ]

Population Order: Normal

User Options :

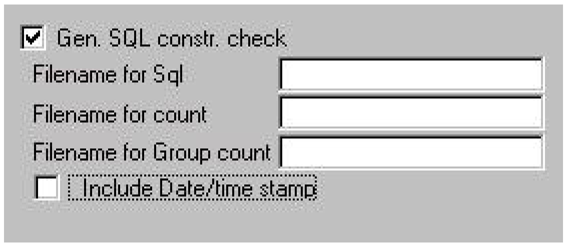
Property	Value
TABLESPACE	BOB_INDEX
STORAGE	INITIAL 1M NEXT 1M

This creates the following:

CREATE INDEX IND$CLI_FK_CLI_GBL ON CLIENT (LAND_ID ASC) TABLESPACE BOB_INDEX
STORAGE(INITIAL 1M NEXT 1M);

2.9 Creating a Foreign Keys SQL Check

You can generate a script that checks if your database contents is in accordance with your constraints. This is useful if you add constraints to the database, or if constraints where not previously applied and you want to do a database audit. Check '**Gen. SQL constr. check**' on the **constraints** tab.



✓ Gen. SQL constr. check	
Filename for Sql	
Filename for count	
Filename for Group count	
☐ Include Date/time stamp	

In a user relation you can add the user option: CREATECONSTRAINTS. Setting the value to NO does not generate the specific foreign key constraint.
You can indicate different files for different purposes. In the first file (filename for SQL) a check is created of all foreign keys that do not have matching primary keys. The statements appear like the following:

```
SELECT ADVIES.C_ADVIES,ADVIES.C_SRT_STATUS FROM ADVIES WHERE NOT
EXISTS ( SELECT SOORT_STATUS.C_SRT_STATUS
    FROM SOORT_STATUS WHERE
ADVIES.C_SRT_STATUS=SOORT_STATUS.C_SRT_STATUS );
```

In the second file, a count is made of the number of records that do not conform to the foreign / primary key check. This appears as the following:

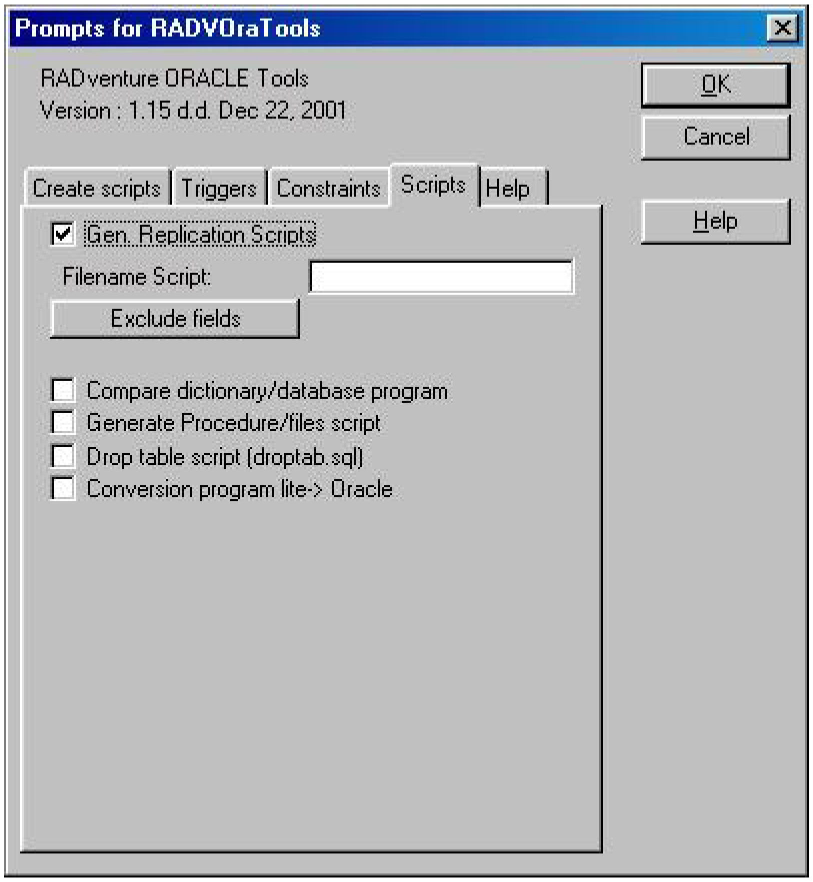
```
SELECT COUNT(*) FROM ADVIES WHERE NOT EXISTS
(SELECT SOORT_STATUS.C_SRT_STATUS
FROM SOORT_STATUS WHERE
ADVIES.C_SRT_STATUS=SOORT_STATUS.C_SRT_STATUS );
```

In the third file, a grouping is made:

SELECT EIND_DIAGNOSE.C_SRT_STATUS,Count(*) FROM EIND_DIAGNOSE
WHERE NOT EXISTS (SELECT SOORT_STATUS.C_SRT_STATUS
FROM SOORT_STATUS WHERE
EIND_DIAGNOSE.C_SRT_STATUS=SOORT_STATUS.C_SRT_STATUS)
GROUP BY EIND_DIAGNOSE.C_SRT_STATUS;

2.10 Creating Various Scripts

With the **Scripts tab** you can generate some useful scripts. This tab appears as follows:


Prompts for RADVOraTools

RADventure ORACLE Tools
Version : 1.15 d.d. Dec 22, 2001

Tabs: Create scripts | Triggers | Constraints | Scripts | Help

- [✓] Gen. Replication Scripts
- Filename Script:
- Exclude fields
- [ ] Compare dictionary/database program
- [ ] Generate Procedure/files script
- [ ] Drop table script (droptab.sql)
- [ ] Conversion program lite-> Oracle

2.11 Creating a Drop Table Script

You can create a drop table script to quickly drop your tables in your users schema. On the **Scripts** tab check '**Drop table script'.** This utility makes a drop table statement for every table in you data dictionary. Statements appear as follows.

DROP TABLE MEMO_CATEGORIE CASCADE CONSTRAINTS;
DROP TABLE ROL CASCADE CONSTRAINTS;

The statements are put in a fixed file named droptab.sql.

2.12 Creating a Dictionary Compare Program

You can create a dictionary compare program by clicking on the **scripts** tab and checking the 'Compare Dictionary/Database Program' option. This generates a Project file Compdict.prj and a Compdict.clw program. You can open the project in Clarion and compile/run the program. After a Logon screen, a report is sent directly to your printer showing the differences between your dictionary and your Oracle tables.

3 Frequently Asked Questions

Question: Can I reuse the table script after I modify my tables.
Answer: Yes, you can, but you will have to drop the tables before recreating them. Using ALTER TABLE on your modifications would be much better. Also The RADventure File class can be used to alter your tables automatically.

4 Version History

Version 1.0 - 1.14 / these version were used for internal purposes only at RADventure. Many enhancements were made to come to the product as it is today.
Version 1.15 / First commercial available release

5 RADventure Support

RADventure Support	
Email:	tools@radventure.nl
Telephone:	+31 (0)346 29 09 80
Fax:	+31 (0)346 29 09 05
Post:	PO Box 1069, 3600 BB Maarssen, The Netherlands

6 Copyright