78

# DBMS_DICTIONARY_CHECK

DBMS_DICTIONARY_CHECK is a read-only and lightweight PL/SQL package procedure that helps you identify Oracle Database dictionary inconsistencies.

## Overview of Oracle Database Dictionary Check

DBMS_DICTIONARY_CHECK is a read-only and lightweight PL/SQL package procedure that helps you identify Oracle Database dictionary inconsistencies that are manifested in unexpected entries in the Oracle Database dictionary tables or invalid references between dictionary tables. Oracle Database dictionary inconsistencies can cause process failures and, in some cases, instance crash. Such inconsistencies may be exposed to internal ORA-00600 errors. DBMS_DICTIONARY_CHECK assists you in identifying such inconsistencies and in some cases provides guided remediation to resolve the problem and avoid such database failures.

Unexpected entries in the dictionary tables or invalid references between dictionary tables, for example, include the following:

- A lob segment not in OBJ$
- An entry in SOURCE$ not in OBJ$
- Invalid data between OBJ$-PARTOBJ$ and TABPART$
- A segment with no owner
- A materialized segment with no entry in seg$
- A segment with no object entry
- A recycle bin object not in the recyclebin$
- Check if Control Seq is near the limit

## Using DBMS_DICTIONARY_CHECK

To run all the checks or only the critical checks defined by DBMS_DICTIONARY_CHECK, connect to the SYS schema, and then run the following commands as SYS user:

**Full check**

```
SQL> set serveroutput on size unlimited
SQL> execute dbms_dictionary_check.full
```

```
SQL> set serveroutput on size unlimited
SQL> EXECUTE dbms_dictionary_check.full(repair=>TRUE)
```

While running a full check, optionally, you can use the repair option to resolve inconsistencies. Valid values: TRUE|FALSE. Default: FALSE.

### Critical check

```
SQL> set serveroutput on size unlimited
SQL> execute dbms_dictionary_check.critical
```

Optionally, turn on the spool to redirect the output to a server-side flat file. By default, when you query the `SYS` schema, the `DBMS_DICTIONARY_CHECK` package creates a trace file named, `DICTCHECK.trc`.

For example: `/<path>/diag/rdbms/<db_name>/<oracle_sid>/trace/<oracle_sid>_<ora>_<pid>_DICTCHECK.trc`.

The execution reports the result as:

- **CRITICAL**: Requires an immediate fix.
- **FAIL**: Requires resolution on priority.
- **WARN**: Good to resolve.
- **PASS**: No issues.

> **Note:**
>
> In all cases, any output reporting "problems" must be triaged by Oracle Support to confirm if any action is required.

### Example 78-1 Full check run

```
SQL> set serveroutput on size unlimited
SQL> execute dbms_dictionary_check.full
dbms_dictionary_check on 07-MAR-2023
03:17:48
-------------------------------------------

Catalog Version 21.0.0.0.0
(2300000000)
db_name:
ORCL

Is CDB?:
NO

Trace File: /oracle/log/diag/rdbms/orcl/orcl/trace/
orcl_ora_2574906_DICTCHECK.trc

                                  Catalog
Fixed
Procedure Name                    Version    Vs Release    Timestamp
Result
------------------------------ ... ---------- -- ---------- --------------
------
.- OIDOnObjCol                 ... 2300000000 <=  *All Rel* 03/07 03:17:48
PASS
```

```
.- LobNotInObj                 ... 2300000000 <=  *All Rel* 03/07 03:17:48
PASS
.- SourceNotInObj              ... 2300000000 <=  *All Rel* 03/07 03:17:48
PASS
.- OversizedFiles              ... 2300000000 <=  *All Rel* 03/07 03:17:48
PASS
.- PoorDefaultStorage          ... 2300000000 <=  *All Rel* 03/07 03:17:48
PASS
.- PoorStorage                 ... 2300000000 <=  *All Rel* 03/07 03:17:48
PASS
.- TabPartCountMismatch        ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- TabComPartObj               ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- Mview                       ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- ValidDir                    ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- DuplicateDataobj            ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- ObjSyn                      ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- ObjSeq                      ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- UndoSeg                     ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- IndexSeg                    ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- IndexPartitionSeg           ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- IndexSubPartitionSeg        ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- TableSeg                    ... 2300000000 <=  *All Rel* 03/07 03:17:49
FAIL

HCKE-0019: Orphaned TAB$ (no SEG$) (Doc ID
1360889.1)
ORPHAN TAB$: OBJ#=83241 DOBJ#=83241 TS=5 RFILE/BLOCK=5/11 TABLE=SYS.ORPHANSEG
BOBJ#=

.- TablePartitionSeg           ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- TableSubPartitionSeg        ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- PartCol                     ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- ValidSeg                    ... 2300000000 <=  *All Rel* 03/07 03:17:49
FAIL

HCKE-0023: Orphaned SEG$ Entry (Doc ID
1360934.1)
ORPHAN SEG$: SegType=LOB TS=5 RFILE/
BLOCK=5/26

.- IndPartObj                  ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
```

```
.- DuplicateBlockUse            ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- FetUet                       ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- Uet0Check                    ... 2300000000 <=  *All Rel* 03/07 03:17:49
PASS
.- SeglessUET                   ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ValidInd                     ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ValidTab                     ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- IcolDepCnt                   ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ObjIndDobj                   ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- TrgAfterUpgrade              ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ObjType0                     ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ValidOwner                   ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- StmtAuditOnCommit            ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- PublicObjects                ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- SegFreelist                  ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ValidDepends                 ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- CheckDual                    ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ObjectNames                  ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ChkIotTs                     ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- NoSegmentIndex               ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- NextObject                   ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- DroppedROTS                  ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- FilBlkZero                   ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- DbmsSchemaCopy               ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- IdnseqObj                    ... 2300000000 >  1201000000 03/07 03:17:50
PASS
.- IdnseqSeq                    ... 2300000000 >  1201000000 03/07 03:17:50
PASS
.- ObjError                     ... 2300000000 >  1102000000 03/07 03:17:50
PASS
.- ObjNotLob                    ... 2300000000 <=  *All Rel* 03/07 03:17:50
FAIL

HCKE-0049: OBJ$ LOB entry has no LOB$ or LOBFRAG$ entry (Doc ID
```

```
2125104.1)
OBJ$ LOB has no LOB$ entry: Obj=83243 Owner: SYS LOB Name:
LOBC1

.- MaxControlfSeq              ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- SegNotInDeferredStg         ... 2300000000 >  1102000000 03/07 03:17:50
PASS
.- SystemNotRfile1             ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- DictOwnNonDefaultSYSTEM     ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ValidateTrigger             ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- ObjNotTrigger               ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
.- InvalidTSMaxSCN             ... 2300000000 >  1202000000 03/07 03:17:50
CRITICAL

HCKE-0054: TS$ has Tablespace with invalid Maximum SCN (Doc ID
1360208.1)
TS$ has Tablespace with invalid Maximum SCN: TS#=5 Tablespace=HCHECK
Online$=1

.- OBJRecycleBin               ... 2300000000 <=  *All Rel* 03/07 03:17:50
PASS
----------------------------------------

07-MAR-2023 03:17:50  Elapsed: 2
secs
----------------------------------------

Found 4 potential problem(s) and 0
warning(s)
Found 1 CRITICAL problem(s) needing
attention
Contact Oracle Support with the output and trace
file
to check if the above needs attention or
not
BEGIN dbms_dictionary_check.full; END;

*
ERROR at line 1:
ORA-20000: dbms_dictionary_check found 1 critical issue(s).  Trace file:
/oracle/log/diag/rdbms/orcl/orcl/trace/orcl_ora_2574906_DICTCHECK.trc

SQL>
```

**Example 78-2 Full check run with repair option**

```
SQL> set serveroutput on size unlimited
SQL> EXECUTE dbms_dictionary_check.full(repair=>TRUE)
dbms_dictionary_check on 04-OCT-2023 01:35:37
-----------------------------------------------
```

```
Catalog Version 23.0.0.0.0 (2300000000)
db_name: orcl
Is CDB?: NO
Trace File: /oracle/log/diag/rdbms/orcl/orcl/trace/
orcl_ora_3831454_DICTCHECK.trc
                                    Catalog       Fixed
Procedure Name                      Version    Vs Release    Timestamp
Result
------------------------------ ... ---------- -- ---------- --------------
------
.- OIDOnObjCol                 ... 2300000000 <=  *All Rel* 10/04 01:35:37
PASS
.- LobNotInObj                 ... 2300000000 <=  *All Rel* 10/04 01:35:37
PASS
.- SourceNotInObj              ... 2300000000 <=  *All Rel* 10/04 01:35:38
FAIL

HCKE-0003: SOURCE$ for OBJ# not in OBJ$ (Doc ID 1360233.1)
SOURCE$ has 10 rows for 1 OBJ# values not in OBJ$


INCONSISTENCY REPAIRED - Check the trace file for repair details:
SourceNotInObj_Repair: DELETED 10 objects from SOURCE$ not found in OBJ$

.- OversizedFiles              ... 2300000000 <=  *All Rel* 10/04 01:35:38
PASS
.- PoorDefaultStorage          ... 2300000000 <=  *All Rel* 10/04 01:35:38
PASS
.- PoorStorage                 ... 2300000000 <=  *All Rel* 10/04 01:35:38
PASS
.- TabPartCountMismatch        ... 2300000000 <=  *All Rel* 10/04 01:35:38
PASS
.- TabComPartObj               ... 2300000000 <=  *All Rel* 10/04 01:35:38
PASS
.- Mview                       ... 2300000000 <=  *All Rel* 10/04 01:35:38
PASS
.- ValidDir                    ... 2300000000 <=  *All Rel* 10/04 01:35:38
PASS
.- DuplicateDataobj            ... 2300000000 <=  *All Rel* 10/04 01:35:38
PASS
.- ObjSyn                      ... 2300000000 <=  *All Rel* 10/04 01:35:39
PASS
.- ObjSeq                      ... 2300000000 <=  *All Rel* 10/04 01:35:39
PASS
.- ValidateSeg                 ... 2300000000 <=  *All Rel* 10/04 01:35:39
FAIL

HCKE-0023: Orphaned SEG$ Entry (Doc ID 1360934.1)
ORPHAN SEG$: SegType=DATA TS=5 RFILE/BLOCK=5/11 HWMINCR(DOBJ#)=73271

.- TableSeg                    ... 2300000000 <=  *All Rel* 10/04 01:35:39
PASS
.- TablePartitionSeg           ... 2300000000 <=  *All Rel* 10/04 01:35:39
PASS
.- TableSubPartitionSeg        ... 2300000000 <=  *All Rel* 10/04 01:35:39
PASS
```

```
ORPHAN SEG$: SegType=DATA TS=5 RFILE/BLOCK=5/11
^ Segment entry repaired - Converted to TEMPORARY
INCONSISTENCY REPAIRED - Check the trace file for repair details:
ValidateSeg repaired 1 Orphan Seg$ entries

.- UndoSeg                    ... 2300000000 <=  *All Rel* 10/04 01:35:39
PASS
...
----------------------------------------
04-OCT-2023 01:35:40  Elapsed: 3 secs
----------------------------------------
Found 3 potential problem(s) and 0 warning(s)
Repaired 11 item(s)
Contact Oracle Support with the output and trace file
to check if the above needs attention or not
BEGIN dbms_dictionary_check.full(repair=>TRUE); END;

*
ERROR at line 1:
ORA-20001: dbms_dictionary_check found 3 problem(s) and repaired 11 item(s).
Trace file:
/oracle/log/diag/rdbms/orcl/orcl/trace/orcl_ora_3831454_DICTCHECK.trc
```

### Example 78-3 Critical check run

```
SQL> set serveroutput on size unlimited
SQL> execute dbms_dictionary_check.critical
dbms_dictionary_check on 07-MAR-2023
03:12:23
--------------------------------------------

Catalog Version 21.0.0.0.0
(2100000000)
db_name:
ORCL

Is CDB?:
NO

Trace File: /oracle/log/diag/rdbms/orcl/orcl/trace/
orcl_ora_2574058_DICTCHECK.trc

                                  Catalog
Fixed
Procedure Name                    Version    Vs Release    Timestamp
Result
------------------------------ ... ---------- -- ---------- --------------
------
.- UndoSeg                    ... 2300000000 <=  *All Rel* 03/07 03:12:23
PASS
.- MaxControlfSeq             ... 2300000000 <=  *All Rel* 03/07 03:12:23
PASS
.- InvalidTSMaxSCN            ... 2300000000 >  1202000000 03/07 03:12:23
CRITICAL
```

```
HCKE-0054: TS$ has Tablespace with invalid Maximum SCN (Doc ID
1360208.1)
TS$ has Tablespace with invalid Maximum SCN: TS#=5 Tablespace=HCHECK
Online$=1

------------------------------------

07-MAR-2023 03:12:23  Elapsed: 0
secs
------------------------------------

Found 1 potential problem(s) and 0
warning(s)
Found 1 CRITICAL problem(s) needing
attention
Contact Oracle Support with the output and trace
file
to check if the above needs attention or
not
BEGIN dbms_dictionary_check.critical; END;

*
ERROR at line 1:
ORA-20000: dbms_dictionary_check found 1 critical issue(s).  Trace file:
/oracle/log/diag/rdbms/orcl/orcl/trace/orcl_ora_2574058_DICTCHECK.trc

SQL>
```

# Summary of DBMS_DICTIONARY_CHECK Subprograms

`DBMS_DICTIONARY_CHECK` package includes the following procedures:

- **LobNotInObj**: Checks if a LOB segment is not in `OBJ$` (My Oracle Support Note 1360208.1)
- **OIDOnObjCol**: Checks if an object type column is not in `OID$` (My Oracle Support Note 1360268.1)
- **SourceNotInObj**: Checks if an entry in `SOURCE$` is not in `OBJ$` (My Oracle Support Note 1360233.1)
  While running the `SourceNotInObj` procedure, optionally, you can use the `repair` option to resolve inconsistencies. Valid values: `TRUE|FALSE`. Default: `FALSE`.
- **IndIndparMismatch**: Checks for index name mismatch between partitions (My Oracle Support Note 1360285.1)
- **InvCorrAudit**: Checks for invalid `AUDIT$` entries (My Oracle Support Note 1360489.1)
- **OversizedFiles**: Checks for oversized database files (My Oracle Support Note 1360490.1)
- **PoorDefaultStorage**: Checks tablespace default storage clauses (My Oracle Support Note 1360493.1)
- **PoorStorage**: Checks objects storage clause (My Oracle Support Note 1360496.1)
- **PartSubPartMismatch**: Checks valid partition methods (My Oracle Support Note 1360504.1)

- **TabPartCountMismatch**: Checks invalid data between `OBJ$-PARTOBJ$` and `TABPART$` (My Oracle Support Note 1360514.1)
- **TabComPartObj**: Checks that the composite partition has a valid entry in `OBJ$` (My Oracle Support Note 1360515.1)
- **Mview**: Check invalid entries for materialized view (My Oracle Support Note 1360517.1)
- **ValidDir**: Checks that the directory object has valid entries (My Oracle Support Note 1360518.1)
- **DuplicateDataobj**: Checks for duplicate segment `data_object_id` (My Oracle Support Note 1360519.1)
- **ObjSyn**: Checks that a synonym has a valid entry in `OBJ$` (My Oracle Support Note 1360520.1)
- **ObjSeq**: Checks that a sequence has a valid entry in `OBJ$` (My Oracle Support Note 1360524.1)
- **UndoSeg**: Checks that undo segment has a valid entry in `SEG$` (My Oracle Support Note 1360527.1)
- **IndexSeg**: Checks that an index segment has a valid entry in `SEG$` (My Oracle Support Note 1360531.1)
- **IndexPartitionSeg**: Checks that an index partition has a valid entry in `SEG$` (My Oracle Support Note 1360535.1)
- **IndexSubPartitionSeg**: Checks that an index sub-partition has a valid entry in `SEG$` (My Oracle Support Note 1360536.1)
- **TableSeg**: Checks that a table has a valid entry in `SEG$` (My Oracle Support Note 1360889.1)
- **TablePartitionSeg**: Checks that a table partition has a valid entry in `SEG$` (My Oracle Support Note 1360890.1)
- **TableSubPartitionSeg**: Checks that a table sub-partition has a valid entry in `SEG$` (My Oracle Support Note 1360891.1)
- **PartCol**: Checks for valid entry of column partition (My Oracle Support Note 1360892.1)
- **ValidateSeg**: Checks that a segment in `SEG$` has an entry in its parent (My Oracle Support Note 1360934.1)
  While running the `ValidateSeg` procedure, optionally, you can use the `repair` option to resolve inconsistencies. Valid values: `TRUE|FALSE`. Default: `FALSE`.
- **IndPartObj**: Checks that an index partition has an entry in `OBJ$` (My Oracle Support Note 1360935.1)
- **DuplicateBlockUse**: Checks for a segment header block is used by only one segment (My Oracle Support Note 1360880.1)
- **FetUet**: Checks valid free/used space in a dictionary-managed tablespace (My Oracle Support Note 1360882.1)
- **Uet0Check**: Checks valid first extent in a dictionary-managed tablespace (My Oracle Support Note 1360883.1)
- **ExtentlessSeg**: Checks `SEG$`/`UET$` mismatch in a dictionary-managed tablespace (My Oracle Support Note 1360944.1)
- **SeglessUET**: Checks `UET$`/`SEG$` mismatch in a dictionary-managed tablespace (My Oracle Support Note 1360944.1)

- **ValidInd**: Checks that an index in `OBJ$` has a corresponding entry in the index dictionary (My Oracle Support Note 1360528.1)
- **ValidTab**: Checks that a table in `OBJ$` has a corresponding entry in `TAB$` (My Oracle Support Note 1360538.1)
- **IcolDepCnt**: Checks valid entries in `ICOLDEP$` (My Oracle Support Note 1360938.1)
- **WarnIcolDep**: Checks that an index does not have an ADT (object column) (My Oracle Support Note 1360939.1)
- **ObjIndDobj**: Checks index data_object_id mismatch between `OBJ$` and `IND$` (My Oracle Support Note 1360968.1)
- **TrgAfterUpgrade**: Checks valid entries in triggers (My Oracle Support Note 1361014.1)
- **ObjType0**: Checks that `OBJ$` has a valid type greater than 0 (My Oracle Support Note 1361015.1)
- **ValidOwner**: Checks that an entry in `OBJ$` has a valid user ID (My Oracle Support Note 1361020.1)
- **StmtAuditOnCommit**: Checks valid entries for `STMT_AUDIT_OPTION_MAP` (My Oracle Support Note 1361021.1)
- **PublicObjects**: Checks that objects are not owned by `PUBLIC` (My Oracle Support Note 1361022.1)
- **SegFreelist**: Checks that a LOB segment has a valid free list group (My Oracle Support Note 1361023.1)
- **ValidDepends**: Checks for valid dependency timestamps (My Oracle Support Note 1361045.1)
- **CheckDual**: Checks valid entries in DUAL (My Oracle Support Note 1361046.1)
- **ObjectNames**: Checks if an object has the same name as its schema owner (My Oracle Support Note 2363142.1)
- **CboHiLo**: Checks for valid entries in histograms (My Oracle Support Note 1361047.1)
- **ChkIotTs**: Checks that an IOT object does not have a segment (My Oracle Support Note 1361048.1)
- **NoSegmentIndex**: Checks for `NOSEGMENT` indexes (My Oracle Support Note 1361049.1)
- **NextObject**: Checks for valid `data_object_id` (My Oracle Support Note 2124772.1)
- **DroppedROTS**: Checks for valid entries in a read-only tablespace (My Oracle Support Note 2124774.1)
- **FilBlkZero**: Checks for zero data block address (My Oracle Support Note 2124783.1)
- **DbmsSchemaCopy**: Checks for invalid execution of `DBMS_SCHEMA_COPY` (My Oracle Support Note 2124795.1)
- **IdnseqObj**: Checks that Identity column has a valid object (My Oracle Support Note 2124805.1)
- **IdnseqSeq**: Checks that a sequence has a valid object (My Oracle Support Note 2124787.1)
- **ObjError**: Checks that an object error is valid (My Oracle Support Note 2124788.1)
- **ObjNotLob**: Checks that a LOB object has an entry in `LOB$` (My Oracle Support Note 2125104.1)

- **MaxControlfSeq**: Checks if `Control Seq` is near the limit (My Oracle Support Note 2128446.1)
- **SegNotInDeferredStg**: Checks for an invalid deferred segment (My Oracle Support Note 2298947.1)
- **SystemNotRfile1**: Checks that the system tablespace has a relative file number 1 (My Oracle Support Note 2364065.1)
- **DictOwnNonDefaultSYSTEM**: Checks that the users SYS and SYSTEM have default tablespace SYSTEM (My Oracle Support Note 2377270.1)
- **ValidateTrigger**: Checks that triggers have valid entries in their parents (My Oracle Support Note 2384373.1)
- **ObjNotTrigger**: Checks if an object trigger is not in `TRIGGER$` (My Oracle Support Note 2384392.1)
- **WarningTSMaxSCN**: Checks exposed SCN entries in tablespaces
- **InvalidTSMaxSCN**: Checks for invalid SCN entries in tablespaces (My Oracle Support Note 1360208.1)
- **OBJRecycleBin**: Checks that recycle bin objects in `OBJ$` exist in `RECYCLEBIN$` (My Oracle Support Note 2902943.1)
- **LobSeg**: Checks that a LOB segment has a valid entry in `SEG$` (My Oracle Support Note 2948392.1 and 2948408.1)
- **ObjLogicalConstraints:** Checks logical constraints in OBJ$ (My Oracle Support Note 2977609.1 and Note 2977591.1)
- **SysSequences:** Checks inconsistencies with critical sequences owned by SYS (My Oracle Support Note 2992123.1, Note 2992124.1 and Note 2992107.1)
- **ValidateObjStub:** Check orphaned stub entries in OBJ$ (My Oracle Support Note 3037043.1)
- **Critical**: Executes only critical checks
- **Full**: Executes all checks

**Example 78-4 ValidateSeg with repair option**

```
SQL> EXECUTE dbms_dictionary_check.ValidateSeg(repair=>TRUE)
.- ValidateSeg                 ... 2300000000 <=  *All Rel* 10/04 01:30:21
FAIL

HCKE-0023: Orphaned SEG$ Entry (Doc ID 1360934.1)
ORPHAN SEG$: SegType=DATA TS=5 RFILE/BLOCK=5/11 HWMINCR(DOBJ#)=73270

.- TableSeg                    ... 2300000000 <=  *All Rel* 10/04 01:30:21
PASS
.- TablePartitionSeg           ... 2300000000 <=  *All Rel* 10/04 01:30:21
PASS
.- TableSubPartitionSeg        ... 2300000000 <=  *All Rel* 10/04 01:30:21
PASS
ORPHAN SEG$: SegType=DATA TS=5 RFILE/BLOCK=5/11
^ Segment entry repaired - Converted to TEMPORARY
INCONSISTENCY REPAIRED - Check the trace file for repair details:
ValidateSeg repaired 1 Orphan Seg$ entries
```

```
PL/SQL procedure successfully completed.
```

### Example 78-5 SourceNotInObj with repair option

```
SQL> EXECUTE dbms_dictionary_check.SourceNotInObj(repair=>TRUE)
dbms_dictionary_check on 04-OCT-2023 01:30:20
-----------------------------------------------
Catalog Version 23.0.0.0.0 (2300000000)
db_name: orcl
Is CDB?: NO
Trace File: /oracle/log/diag/rdbms/orcl/orcl/trace/
orcl_ora_3831239_DICTCHECK.trc

                                    Catalog       Fixed
Procedure Name                      Version    Vs Release    Timestamp
Result
------------------------------ ... ---------- -- ---------- --------------
------
.- SourceNotInObj              ... 2300000000 <=  *All Rel* 10/04 01:30:20
FAIL

HCKE-0003: SOURCE$ for OBJ# not in OBJ$ (Doc ID 1360233.1)
SOURCE$ has 10 rows for 1 OBJ# values not in OBJ$


INCONSISTENCY REPAIRED - Check the trace file for repair details:
SourceNotInObj_Repair: DELETED 10 objects from SOURCE$ not found in OBJ$

PL/SQL procedure successfully completed.
```