OASIS Health Check User Guide



Table of Contents

[1 Summary 2](#_Toc528331870)

[1.1 Intended Audience 2](#_Toc528331871)

[2 Installation / Uninstallation 2](#_Toc528331872)

[2.1 Assumptions / Requirements 2](#_Toc528331873)

[2.2 Installation 2](#_Toc528331874)

[2.3 Uninstallation 4](#_Toc528331875)

[3 Tool Usage 5](#_Toc528331876)

[3.1 Command Line Help 5](#_Toc528331877)

[3.2 Run OASIS Health Check from the Command Line 6](#_Toc528331878)

[3.3 Run OASIS Health Check for Multiple Environments 8](#_Toc528331879)

[4 Test Reports 9](#_Toc528331880)

[4.1 Last Run Report 9](#_Toc528331881)

[4.2 List of all tests run with dates and run sequence for the day 10](#_Toc528331882)

[4.3 Report of a Test Run for a Particular date and run 11](#_Toc528331883)

[4.4 All Reports 11](#_Toc528331884)

[5 Test Interpretations & Validations - Advanced Users 12](#_Toc528331885)

[5.1 Health Check Results 12](#_Toc528331886)

[5.2 Health Check Interpretations 13](#_Toc528331887)

[6 Self-Service 16](#_Toc528331888)

[6.1 Self-Service Steps 16](#_Toc528331889)

[6.2 An Example of a Self-Service Procedure Performed in odev20181se 16](#_Toc528331890)

[7 Troubleshooting 19](#_Toc528331891)

[7.1 Possible Errors 19](#_Toc528331892)

[7.2 OHC Test Matrix 20](#_Toc528331893)

[8 Future Enhancements 51](#_Toc528331894)

# Summary

OASIS Health Check tool (OHC) is used to monitor OASIS database environments for inconsistencies in configuration and data. Every time the tool is run, routine checks for validity of the OASIS configuration and data are conducted.

*This tool does not monitor OASIS web applications.*

While this tool can be run multiple times a day, daily health check runs are recommended. The tool creates a report of every run in a table; the report can be downloaded as a comma-separated file.

OASIS Health Check tool can be downloaded [from Delphi FTP site](http://ftp.delphi-tech.com/pub/OASIS/Utilities/Oasis_Health_Check).

## Intended Audience

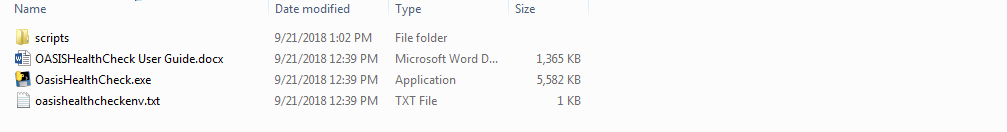
This document is intended to help orient database and system administrators to several database and system maintenance-related tasks and the tools at their disposal.

# Installation / Uninstallation

## Assumptions / Requirements

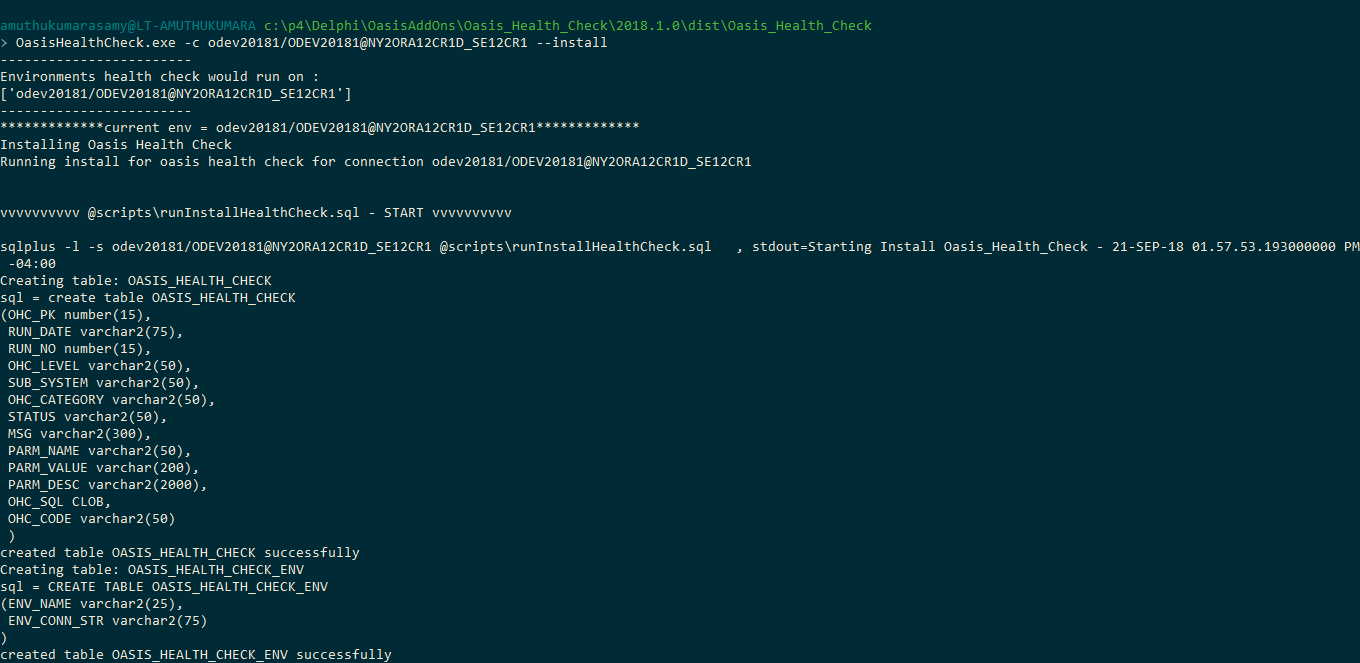
1. Users must be able to log into the target database from the host machine where OHC Tool is run.
2. TNS alias names must be set up on the host machine where OHC is run.

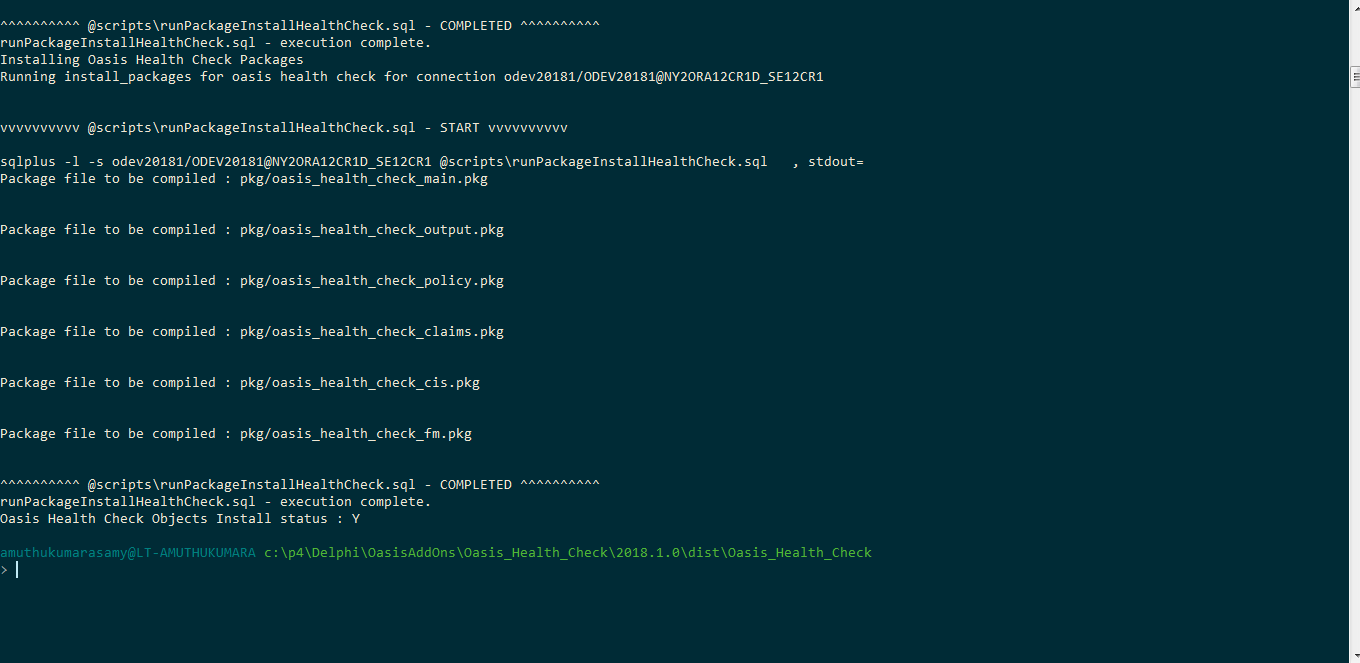
## Installation

1. Download [the zip file](http://ftp.delphi-tech.com/pub/OASIS/Utilities/Oasis_Health_Check) to a preferred directory.
2. Unzip the file. **Oasis\_Health\_Check** folder is created automatically; its contents are shown below:
3. Open the Command Prompt and use the CD command to open the downloaded OASIS Health Check executable directory.
4. Connection string to the database is a required parameter that must be provided with the **-c** command line option and the action as **install**:

* **OasisHealthCheck.exe -c user/passwd@dbhost\_dbschema –-install**

See below for sample installation screenshots:





An installation log is created in the installation directory. New oracle objects such as tables, sequences and packages are installed to the database. Existing objects in the database are not modified due to the tool installation.

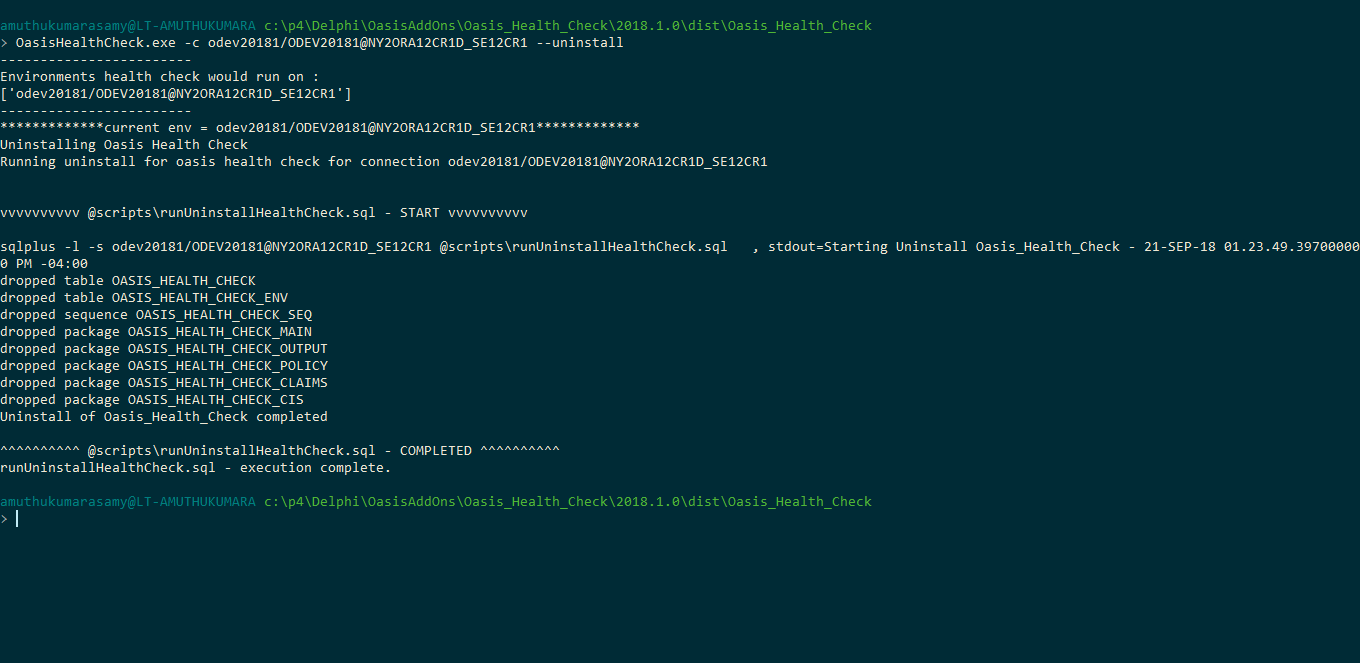
To ensure the installation was successful:

* Check for any errors listed in the output or in the log file.
* Make sure that the **Install status** at the end of the output is set to **‘Y**’.
* If errors had been found and installation was partially completed, follow directions to uninstall OHC in the next section, fix the issues, and try installing again.

## Uninstallation

Run the following command in the command line window to uninstall the OHC tool, as shown below:

* OasisHealthCheck.exe -c user/passwd@dbhost\_dbschema –-uninstall



An uninstall log is created in the root directory. All the new objects such as tables, sequences and packages that were installed during installation is uninstalled. No modifications are done to existing oracle objects.

# Tool Usage

This section describes the three methods of using the OHC tool:

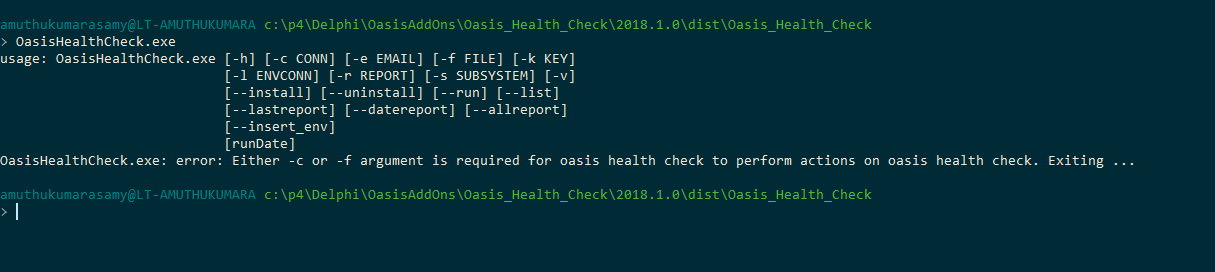
* Using a command line parameter to provide the connection string to the database.
* Provisioning the file with environments to be monitored with OHC.
* Provisioning the database with environments to be monitored with OHC.

At present, provisioning to the file or database uses plain text username and passwords. Next release will include making them more obscure for end users. Until then, access to this file or table (used exclusively to automate the testing) should be limited to personnel with administrative permissions only.

## Command Line Help

To use the basic command line options to run the tool, enter the following command:

* OasisHealthCheck.exe



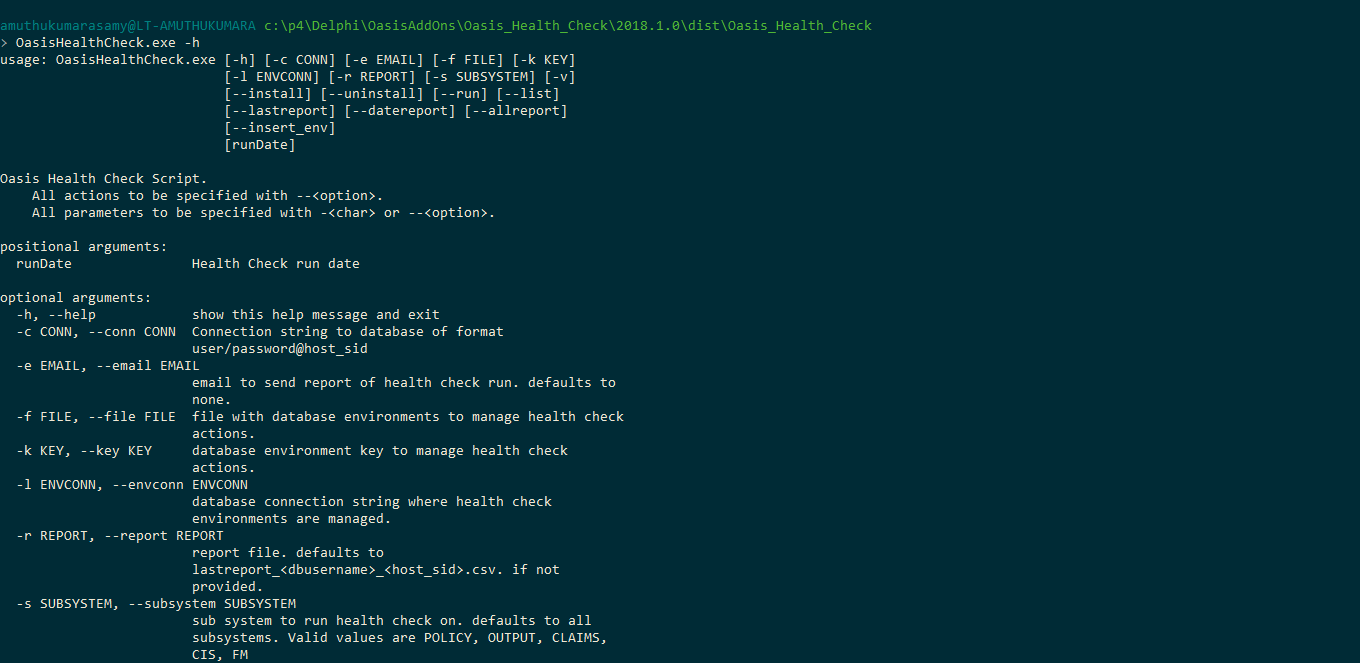
To get the version of the tool that installed,

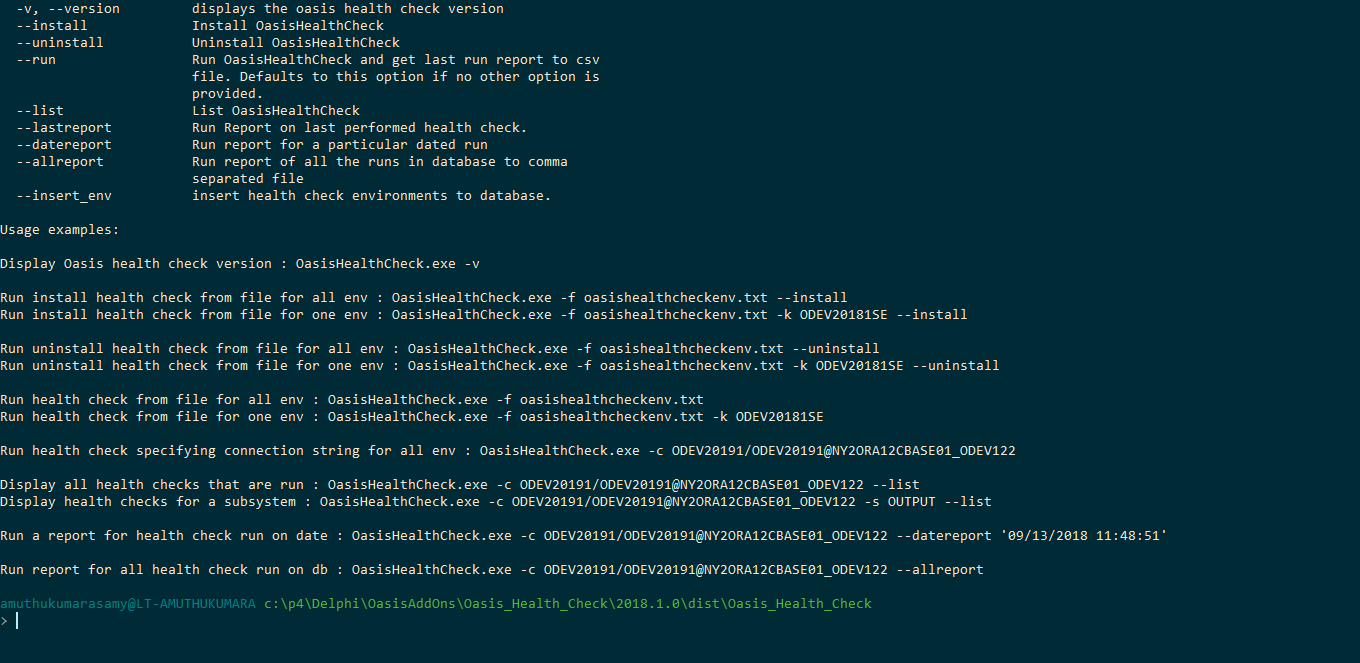
* OasisHealthCheck.exe -v



To get detailed help messages such as usage and details on each option, enter the following command:

* OasisHealthCheck.exe -h (detailed help messages).





## Run OASIS Health Check from the Command Line

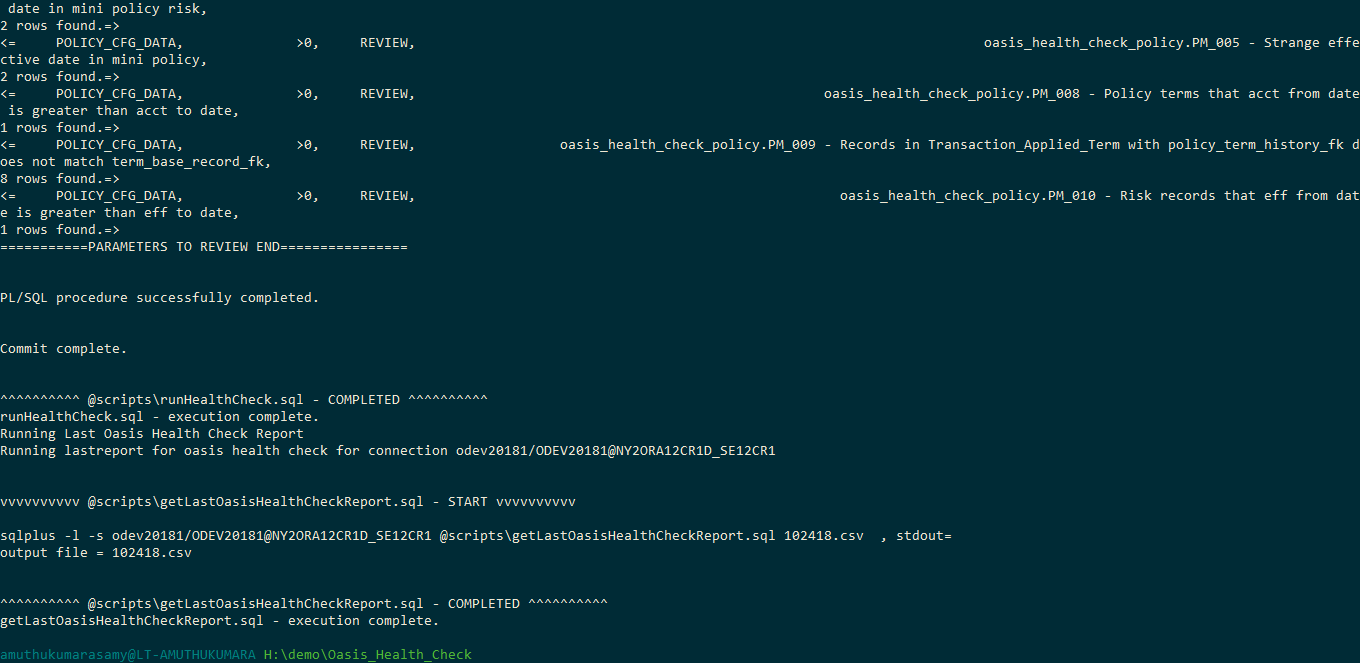
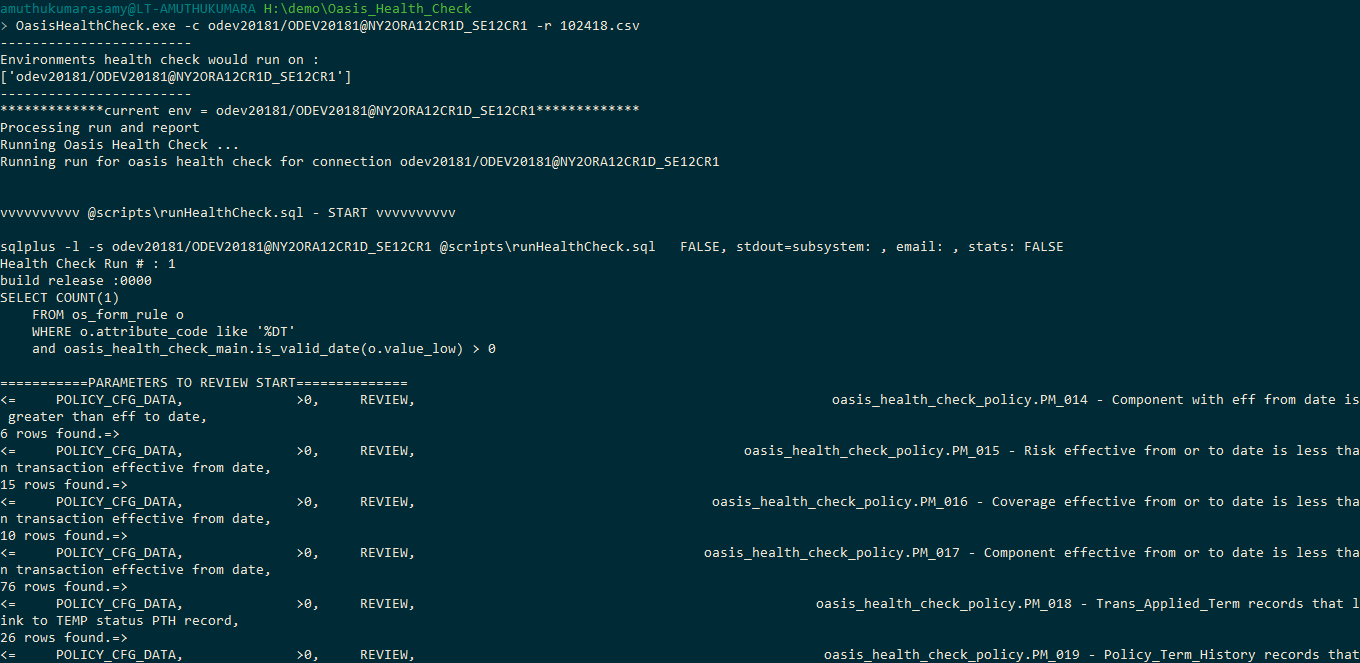
Once OASIS Health Check tool is installed, type in the following command,

* OasisHealthCheck.exe -c user/passwd@dbhost\_dbschema (*defaults to –-run action*)

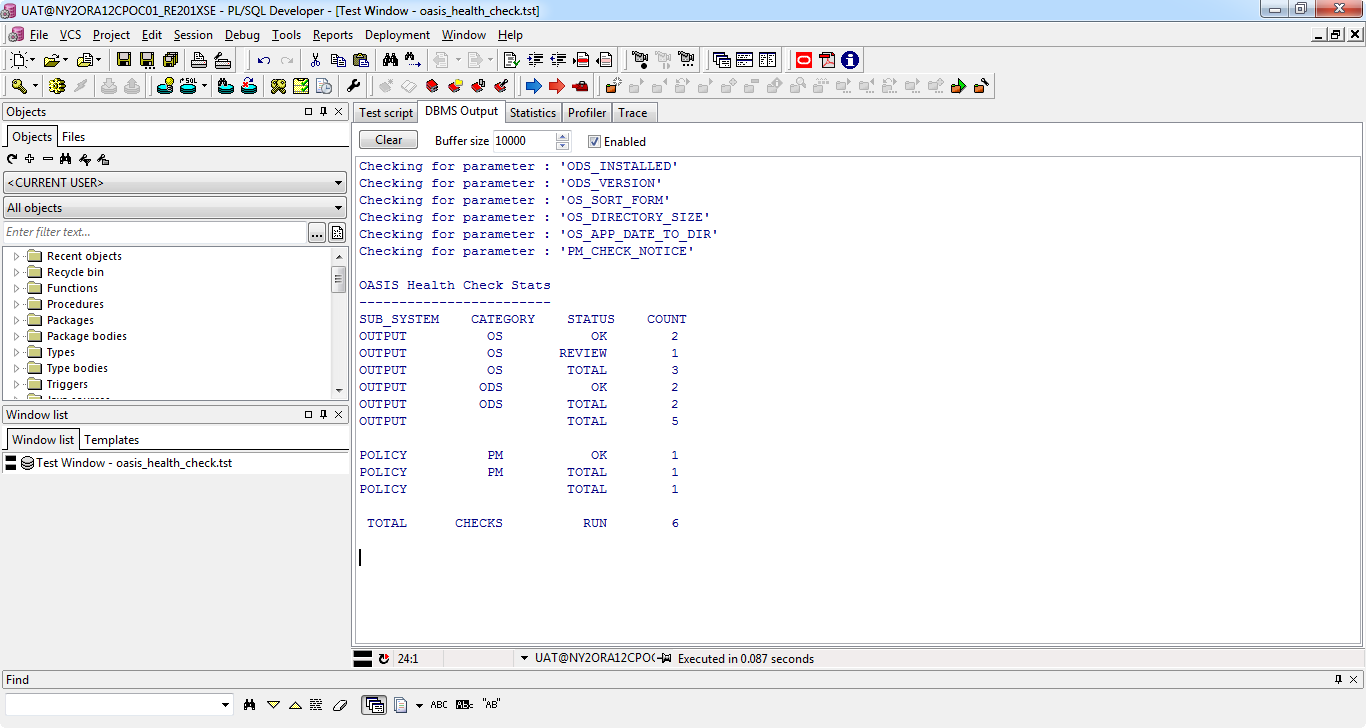
Or

* OasisHealthCheck.exe -c user/passwd@dbhost\_dbschema –-run

OHC defaults to Run action if no action is specified.



A test file included in the **scripts/oasis\_health\_check.tst** directory can be used to run OASIS Health Check tool manually using PL/SQL Developer environment. This method of running a test provides control over the commit of the health check test results. Test runs are not committed to the “OASIS\_HEALTH\_CHECK” table by default unlike when the tool is run using the OHC executable where commit of results is done immediately after execution of tests.



## Run OASIS Health Check for Multiple Environments

### Run Oasis Health Check Using Config File

OASIS Health Check can be run in the environments listed in the following format:

<Key> <Connection String in the format user/password@host\_sid>

Refer to sample “**oasishealthcheckenv.txt”** in the OASIS Health Check tool directory.

After creating the environment file, the following command runs the health check in the environment pointed to by the key:

* OASISHealthCheck.exe -f <filename> -k <key>

If no key is provided, health check runs for all the environments specified in the health check environments file for the command shown below:

* OASISHealthCheck.exe -f <filename>

### Create Database Config Table Data

OASIS Health Check environments can be provisioned in a database table created during installation (OASIS\_HEALTH\_CHECK\_ENV) using a sample script provided in the **scripts** directory:

**scripts/ insertHealthCheckEnv.sql**

To provision the database environments through the OasisHealthCheck.py script, update the SQL in the script shown above for all the environments that OASIS Health Check has to run on, and run the following command:

* OasisHealthCheck.exe -c <connection string of the db where env has to be installed> --insert\_env

### Run OASIS Health Check Using DB Config Table

OASIS Health Check can be run with the environments provisioned in a database table created during installation (**OASIS\_HEALTH\_CHECK\_ENV**) once the step in [the previous section](#_Set_Up_OASIS) is completed.

The following command runs the health check on the environment specified by the key:

* OASISHealthCheck.exe -l <connection string where the environments are maintained> -k <key>

If no key is provided, the health check runs for all the environments specified in the health check environments’ file for the command shown below:

* OASISHealthCheck.exe -f <connection string where the environments are maintained>

# Test Reports

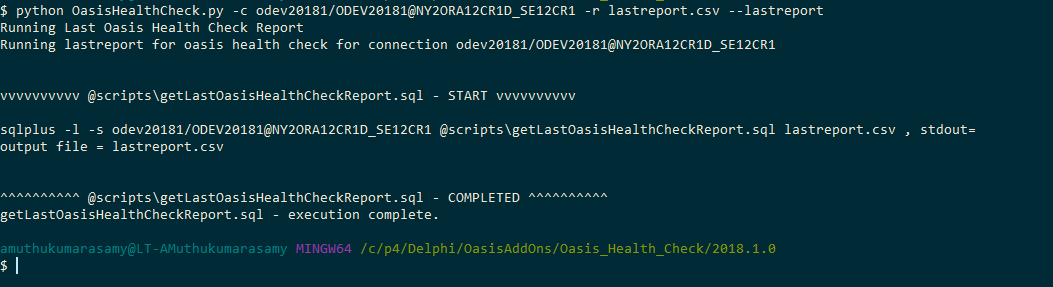
After oasis health check has run successful, OHC tool could be used to download various reports in a comma separated file format that could be opened through Microsoft Excel. Every run will capture that latest run results to a default file or report file specified through -r option.

The various reports that could be run using the tool are described in the sections below.

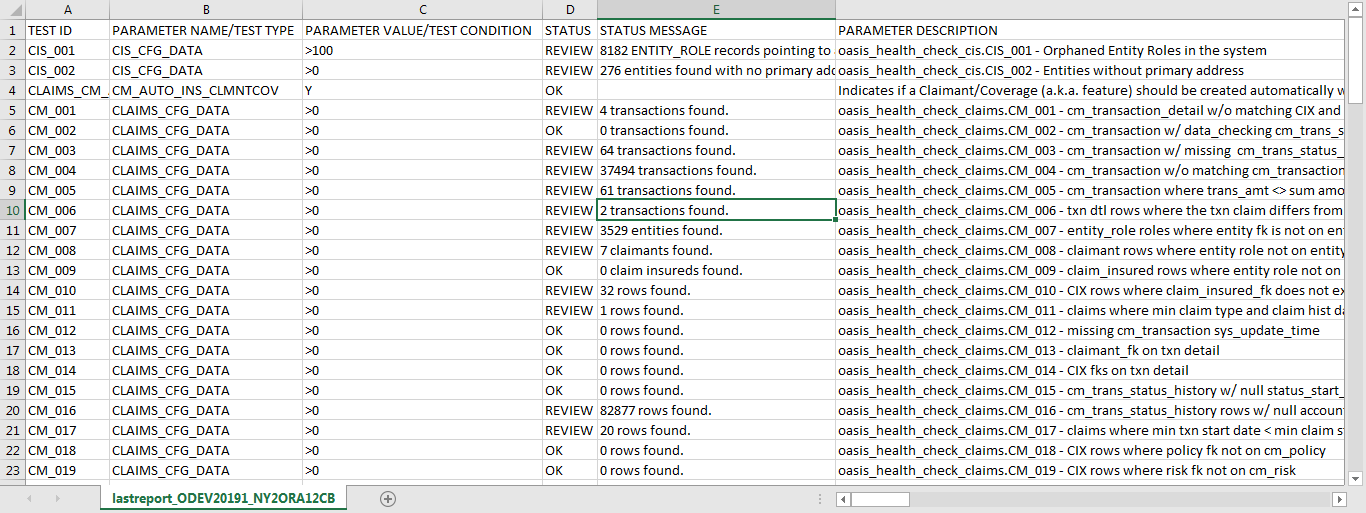
## Last Run Report

To get the last run report in a comma-separated file, type in the following command:

* OasisHealthCheck.exe -c user/passwd@dbhost\_dbschema --lastreport



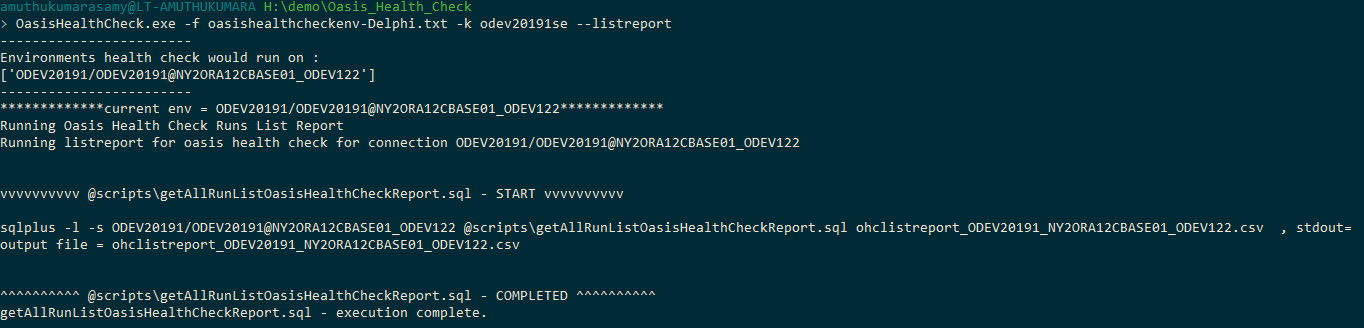
Example of a report:

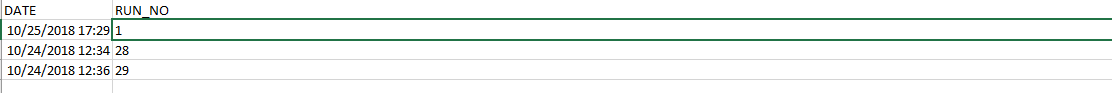


## List of all tests run with dates and run sequence for the day

To get a report for a timestamp in a comma-separated file, type in the following command:

* OasisHealthCheck.exe -c user/passwd@dbhost\_dbschema -r listreport.csv –listreport

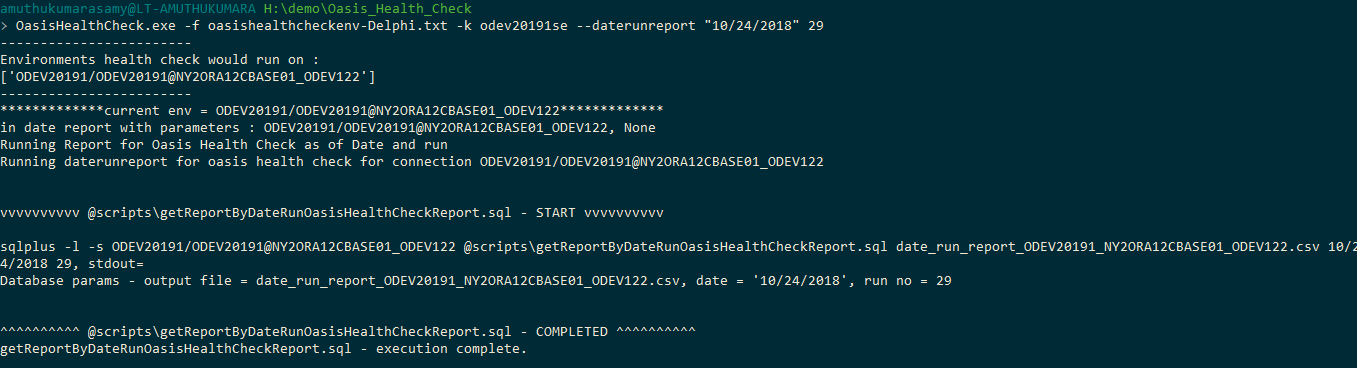
This report would produce a comma separated file as given below,

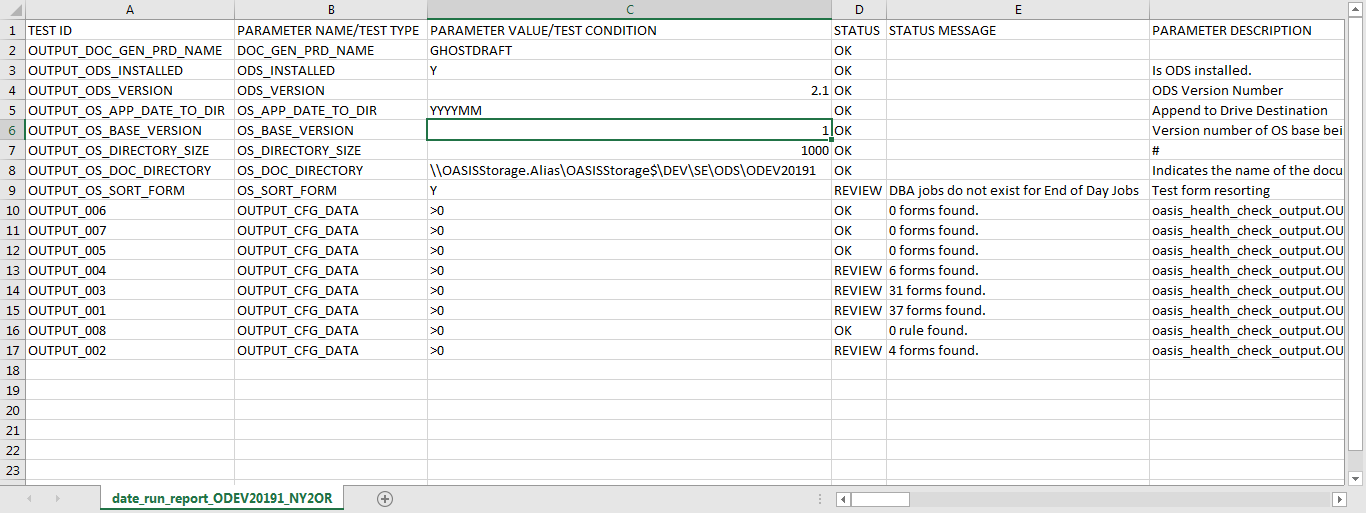


## Report of a Test Run for a Particular date and run

To get a report for a timestamp in a comma-separated file, type in the following command:

* OasisHealthCheck.exe -c user/passwd@dbhost\_dbschema -r daterunreport.csv --daterunreport “10/24/2018” 29

A sample of all report is given below,

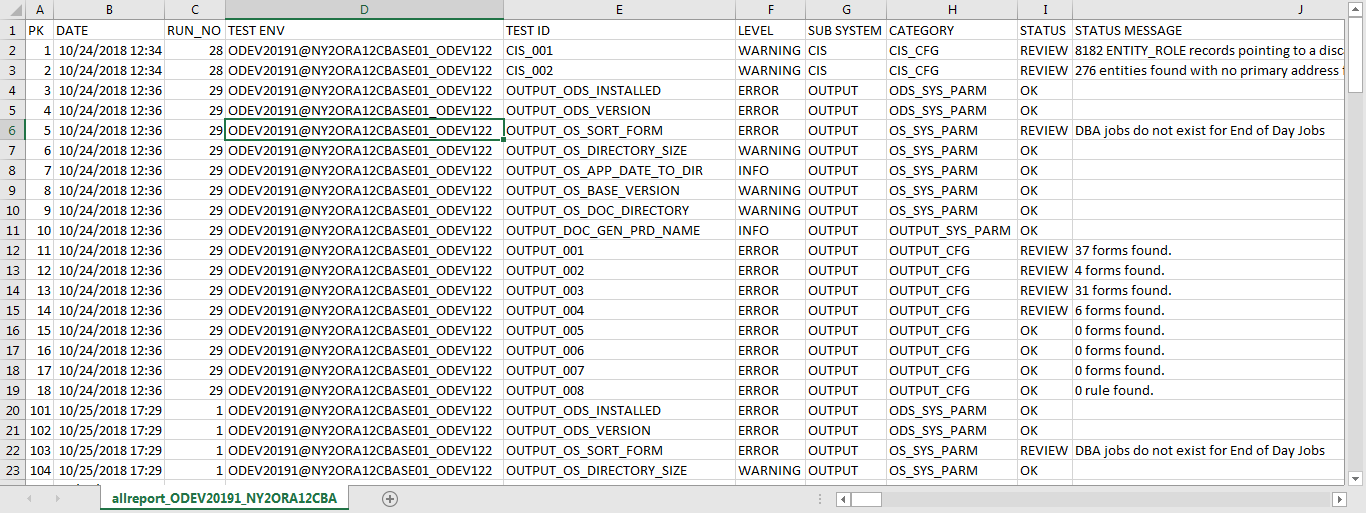


## All Reports

To get all health check reports in the database in a comma-separated file, type in the following command:

* OasisHealthCheck.exe -c user/passwd@dbhost\_dbschema -r allreport.csv –allreport

A sample of all report is given below,



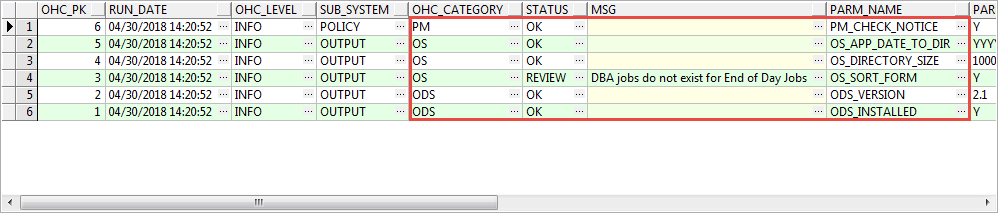
# Test Interpretations & Validations - Advanced Users

This section introduces the location where the health check results are stored, as well as the type of validations performed and how they can be interpreted from the results saved to a table or a comma-separated file.

## Health Check Results

The OHC tool captures the results of the test to the **OASIS\_HEALTH\_CHECK** table. The following information is stored for some of the columns in that table:

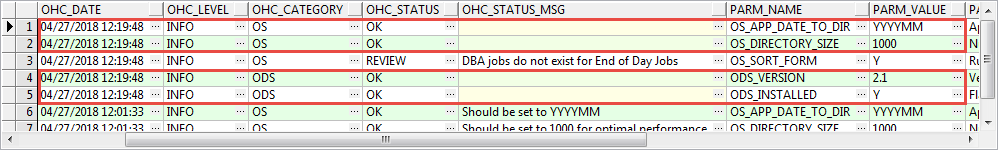
1. **STATUS** column reports the test status with three status codes:
   * OK – Validation was completed and was found to be the expected value.
   * REVIEW – Validation was completed; a configuration issue was found. A status message field specifies the problem for the parameter.
   * PENDING – Validation was not completed when a test case was added without the validation option.
2. Presently, the following subsystems supported by OASIS Health Check tool are shown in the **SUB\_SYSTEM** column:
   * CIS
   * Claims
   * FM
   * Output
   * Policy
3. Two categories of validations are supported as shown in the **OHC\_CATEGORY** column:
   * Configuration - Invalid configuration in the **SYSTEM\_PARAMETER\_UTIL** table for every subsystem. This category is stored in the following format: <SUBSYSTEM>\_SYS\_PARAM.
   * Data inconsistencies – Data inconsistencies found for the subsystem. This category is stored in the following format: <SUBSYSTEM>\_CFG.
4. Validation level is shown using column **OHC\_LEVEL** and is used to assign severity level to the validation. Some of the validation levels are:
   * INFO – Informational. This fix should be applied for OASIS to work efficiently.
   * WARNING – Problem exists, but OASIS is operational.
   * ERROR – The issue must be fixed.



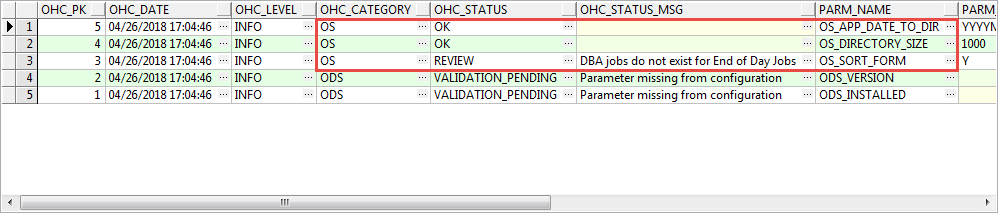
## Health Check Interpretations

This section provides examples of interpreting the OHC tool results. Screenshots below show different test statuses and an explanation of what is expected in that test scenario.

### Test Status – OK and REVIEW



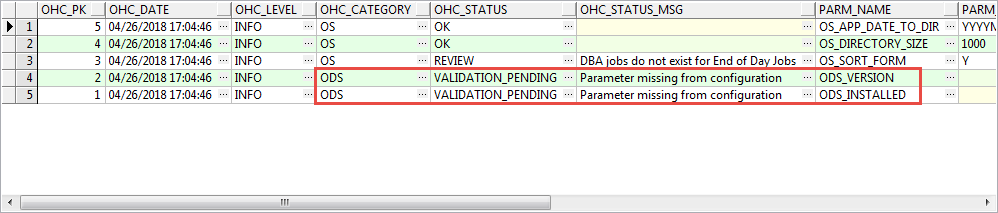
All expected parameter values are displayed with a status of OK and blank status message.



These examples show two types of validations:

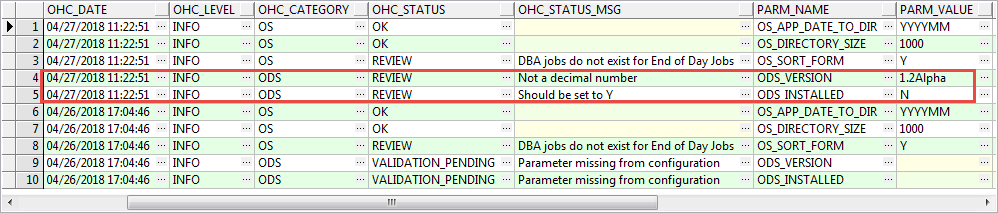
* **OK** status for OS\_APP\_DATE\_TO\_DIR and OS\_DIRECTORY\_SIZE includes a blank status message field.
* **REVIEW** status includes an explanation of which validation failed in the status message field.

### Missing Parameter Configuration



**ODS\_INSTALLED** and **ODS\_VERSION** parameters are not set. Status is set to **Validation\_Pending** since validation was not completed for the missing parameters.

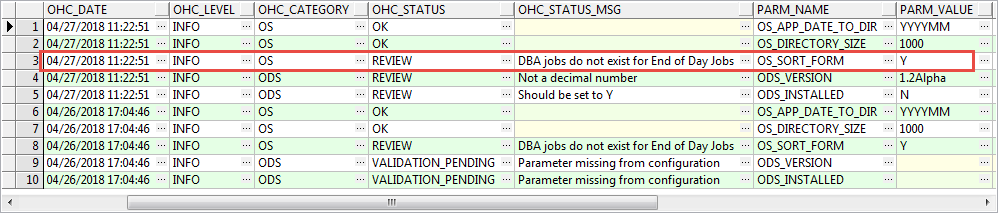
### Invalid Parameter Value



**ODS\_VERSION** is a floating number.

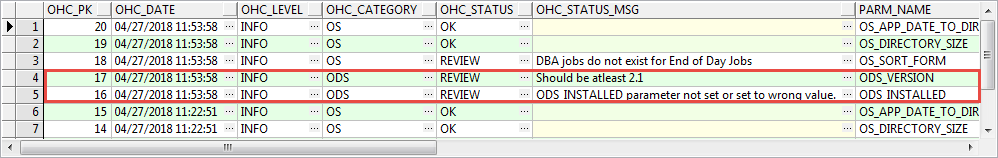
**ODS\_INSTALLED** is expected to be Y.

### Parameter Validation Failure



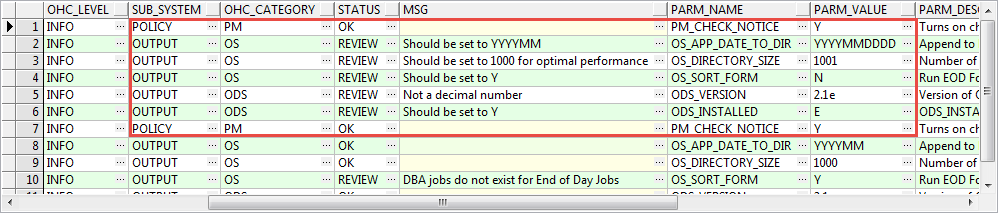
If **OS\_SORT\_FORM** is set to **‘Y’**, jobs must be present to handle EOD for a form that is configured to be **End Of Day**. Since it did not find that job, this is set to **REVIEW**.

### Incorrect Parameter Value



**ODS\_Version** should equal 2.1.

**ODS\_INSTALLED** is expected to be set to **‘Y’.**



**OS\_APP\_DATE\_TO\_DIR** is expected to be set as **‘YYYYMM’.**

**OS\_DIRECTORY\_SIZE** is expected to be set to **1000**.

**OS\_SORT\_FORM** is expected to be set to ‘**Y’.**

**ODS\_VERSION** is expected to be **2.1**.

# Self-Service

Customers may resolve the issues on their own using OHC tool. Below is an example of self-service using one of Delphi environments.

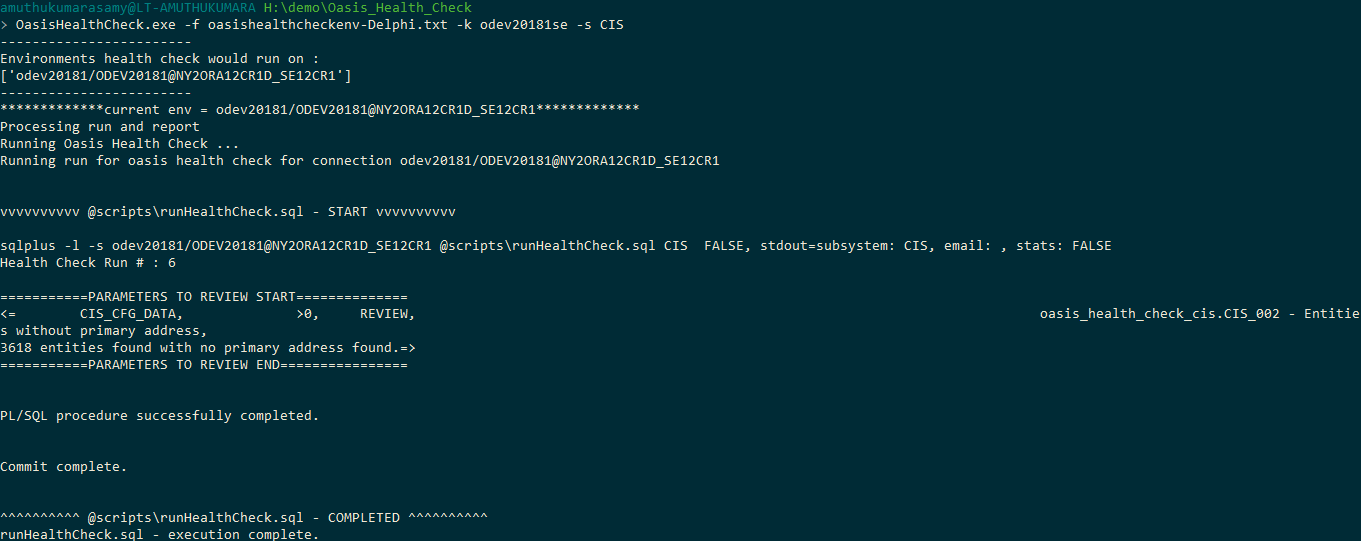
## Self-Service Steps

The following steps are required to successfully resolve an issue found during OASIS health check process:

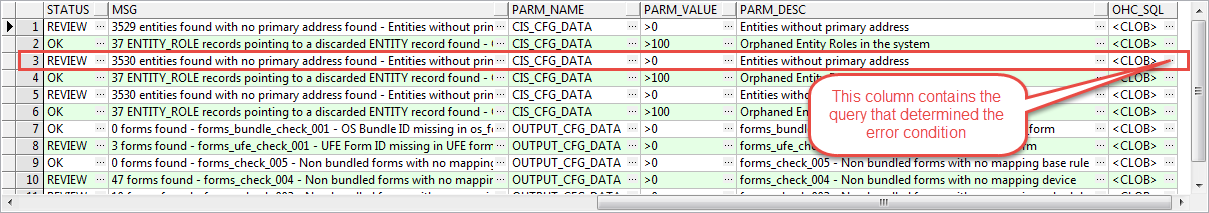
1. Run health check for a small subset (one subsystem).
2. Check for any errors (review the ohc\_sql used to trigger the error).
3. Fix any found errors.
4. Run the same health check for that small subset (one subsystem).
5. Check to see if the errors have been resolved.

## An Example of a Self-Service Procedure Performed in odev20181se

1. Run health check for a small subset (one subsystem, in this case CIS).



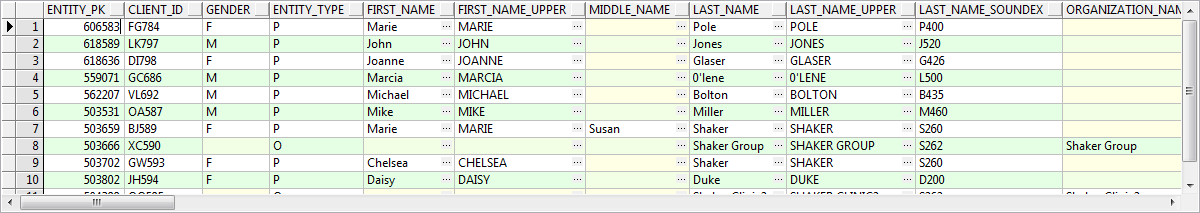
1. Check for any errors (look at the ohc\_sql used to trigger the error).



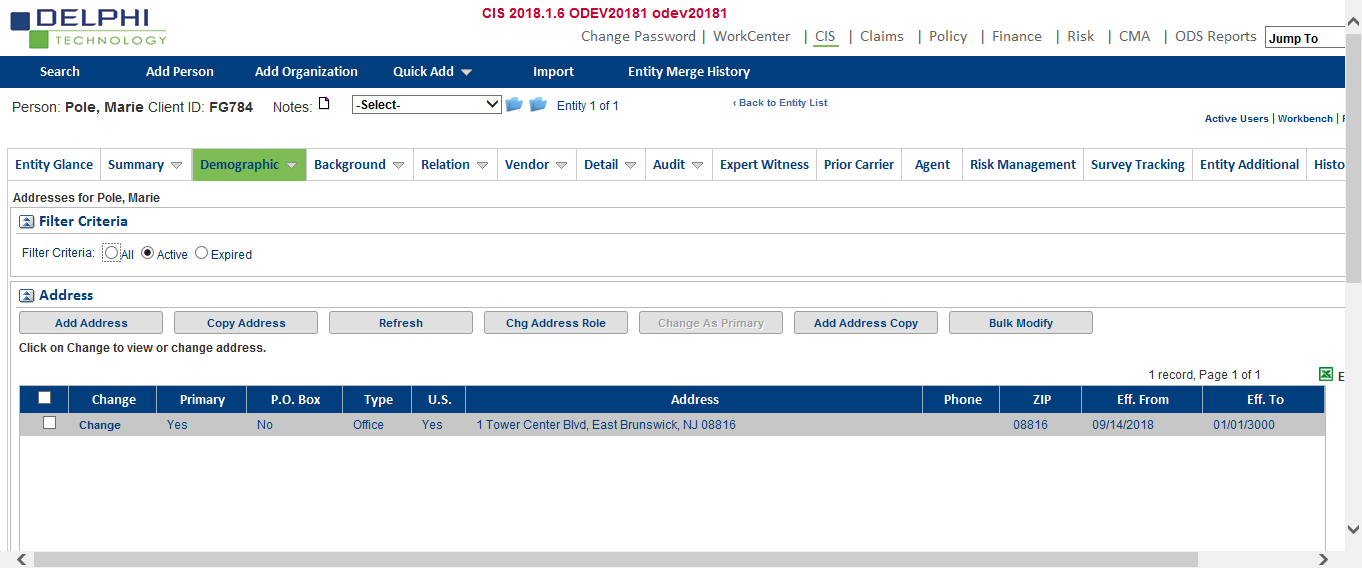
1. Click on the “…” to get the SQL and run the SQL with minor updates as shown below,



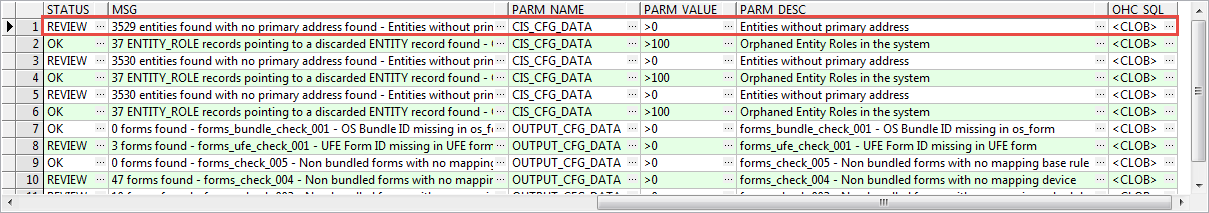
1. The error rows will be displayed



1. Fix the error. In this case, the primary address for the first entity on the query was added.



1. Run the same health check for that small subset (one subsystem, in this case CIS).



1. Check to make sure the error was resolved.

The number of rows of the health check went down by one from 3530 rows on the previous run to 3529 rows on the present run.

# Troubleshooting

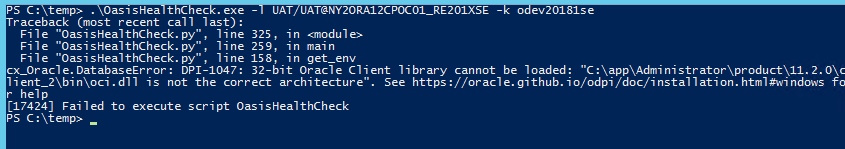
This section lists examples of errors that may be encountered during OASIS health check process and suggested fixes.

## Possible Errors

1. 32-bit Oracle Client Library is not installed.

(To be revisited in a later release.)

OHC tool only works with the Oracle 32-bit client library installed in the host machine. In this case, the host machine had only 64-bit client library installed in the host machine.



1. Excessive number of invalid rows shows up in the health check runs.

If there is an excessive number of records in the OASIS health check test run that show up in REVIEW status **(STATUS** column in **OHC\_HEALTH CHECK** table), customers can refer these results to Delphi by creating a comma separated file of the last report. If a fix is required, Delphi can assist with this effort by creating additional scripts to fix the issue.

## OHC Test Matrix

Tests run with OHC tool are listed in this section with a description and instructions on fixing the issues. This section will be updated as fixes are found for each of the issues. Since this section is bound to evolve, Delphi recommends reviewing this section with each new release.

### TEST ID: CIS\_001

|  |  |
| --- | --- |
| Summary | Check for orphaned Entity Roles in the system. |
| Description | Orphaned ENTITY\_ROLE records pointing to a discarded ENTITY record. |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CIS\_002

|  |  |
| --- | --- |
| Summary | Entities without primary address. |
| Description | Historical Entities that do not have a primary address |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: OUTPUT\_001

|  |  |
| --- | --- |
| Summary | Non-bundled forms with no mapping transaction. |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: OUTPUT\_002

|  |  |
| --- | --- |
| Summary | Non-bundled forms with no mapping copy type. |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: OUTPUT\_003

|  |  |
| --- | --- |
| Summary | Non-bundled forms with no mapping schedule. |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: OUTPUT\_004

|  |  |
| --- | --- |
| Summary | Non-bundled forms with no mapping device. |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: OUTPUT\_005

|  |  |
| --- | --- |
| Summary | Non-bundled forms with no mapping base rule. |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |
|  |  |

### TEST ID: OUTPUT\_006

|  |  |
| --- | --- |
| Summary | OS\_Bundle\_ID missing in OS\_FORM table. |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: OUTPUT\_007

|  |  |
| --- | --- |
| Summary | UFE Form ID missing in UFE form. |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: OUTPUT\_008

|  |  |
| --- | --- |
| Summary | Invalid data format in OS\_FORM\_RULE Table – VALUE\_LOW column. |
| Description | This test would capture any invalid configuration in VALUE\_LOW column of OS\_FORM\_RULE table, if the parameter is a date and is configured with double slashes (10//01/2018) |
| Type | Data Configuration |
| Fix | This script will get any invalid rows configured in the table for the value\_low column.  SELECT \*  FROM os\_form\_rule o  WHERE o.attribute\_code like '%DT'  and oasis\_health\_check\_main.is\_valid\_date(o.value\_low) > 0  ; |

### TEST ID: OUTPUT\_ODS\_INSTALLED

|  |  |
| --- | --- |
| Parameter | ODS\_INSTALLED |
| Description | This parameter should be set to Y if ODS is in the solution architecture. |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | Y or N |

### TEST ID: OUTPUT\_ODS\_VERSION

|  |  |
| --- | --- |
| Parameter | ODS\_VERSION |
| Description | ODS Version number. This can be set to 1 or 2 or any value beyond 2 |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | 2.1 |

### TEST ID: OUTPUT\_OS\_APP\_DATE\_TO\_DIR

|  |  |
| --- | --- |
| Parameter | OS\_APP\_DATE\_TO\_DIR |
| Description | Append to Drive Destination. This has to be set to YYYYMM for optimal performance. It can also be set to YYYYMMDD, but there maybe too many folders to handle. |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | YYYYMM |

### TEST ID: OUTPUT\_DOC\_GEN\_PRD\_NAME

|  |  |
| --- | --- |
| Parameter | DOC\_GEN\_PRD\_NAME |
| Description | This must be set to the third-party forms engine that is a part of the solution architecture. |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | GHOSTDRAFT/ELOQUENCE |

### TEST ID: OUTPUT\_OS\_BASE\_VERSION

|  |  |
| --- | --- |
| Parameter | OS\_BASE\_VERSION |
| Description | Version number of OS base being used |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | 0 for OLD, 1 for NEW, and 2 for both OLD and NEW |

### TEST ID: OUTPUT\_OS\_DIRECTORY\_SIZE

|  |  |
| --- | --- |
| Parameter | OS\_DIRECTORY\_SIZE |
| Description | Number of PDFs that ODS will put into a single folder |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | 1000 |

### TEST ID: OUTPUT\_OS\_DOC\_DIRECTORY

|  |  |
| --- | --- |
| Parameter | OS\_DOC\_DIRECTORY |
| Description | Indicates the name of the document directory. |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | [\\OASISStorage.Alias\OASISStorage$\Dev\SE\ODS\ODEV20181](file://OASISStorage.Alias/OASISStorage$/Dev/SE/ODS/ODEV20181) (Sample) |

### TEST ID: OUTPUT\_OS\_SORT\_FORM

|  |  |
| --- | --- |
| Parameter | OS\_SORT\_FORM |
| Description | Test form resorting |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | Y or N |

### TEST ID: OUTPUT\_OS\_XML\_DIRECTORY

|  |  |
| --- | --- |
| Parameter | OS\_XML\_DIRECTORY |
| Description | The network location to save XML file used by 3rd party ODS form generation product. |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | [\\OASISStorage.Alias\OASISStorage$\Dev\SE\ODS\ODEV20181\XMLDATA](file://OASISStorage.Alias/OASISStorage$/Dev/SE/ODS/ODEV20181/XMLDATA)  (Sample) |

### TEST ID: POLICY\_PM\_CHECK\_NOTICE

|  |  |
| --- | --- |
| Parameter | PM\_CHECK\_NOTICE |
| Description | Turns on check for notices that are stored in PM\_ATTRIBUTE table |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | Y or N |

### TEST ID: PM\_001

|  |  |
| --- | --- |
| Summary | Policy term records that have incorrect CLOSING\_TRANS\_LOG\_FK. |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_002

|  |  |
| --- | --- |
| Summary | TRANSACTION\_APPLIED\_TERM table that has more than one term affected by a transaction but has same POLICY\_TERM\_HISTORY\_FK in TRANSACTION\_APPLIED\_TERM table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_003

|  |  |
| --- | --- |
| Summary | Invalid effective date in mini policy coverage |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_004

|  |  |
| --- | --- |
| Summary | Invalid effective date in mini policy risk |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_005

|  |  |
| --- | --- |
| Summary | Invalid effective date in mini policy |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_006

|  |  |
| --- | --- |
| Summary | Transactions that have timestamp info in ACCOUNTING\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_007

|  |  |
| --- | --- |
| Summary | Policies with more than one primary risk |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_008

|  |  |
| --- | --- |
| Summary | Policy terms with ACCT\_FROM\_DATE greater than ACCT\_TO\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_009

|  |  |
| --- | --- |
| Summary | Records in TRANSACTION\_APPLIED\_TERM table with POLICY\_TERM\_HISTORY\_FK does not match TERM\_BASE\_RECORD\_FK |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_010

|  |  |
| --- | --- |
| Summary | Risks with EFFECTIVE\_FROM\_DATE is greater than EFFECTIVE\_TO\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_011

|  |  |
| --- | --- |
| Summary | Risks with ACCT\_FROM\_DATE is greater than ACCT\_TO\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_012

|  |  |
| --- | --- |
| Summary | Coverage with EFFECTIVE\_FROM\_DATE greater than EFFECTIVE\_TO\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_013

|  |  |
| --- | --- |
| Summary | Coverage with ACCT\_FROM\_DATE is greater than ACCT\_TO\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_014

|  |  |
| --- | --- |
| Summary | Component with EFFECTIVE\_FROM\_DATE greater than EFFECTIVE\_TO\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_015

|  |  |
| --- | --- |
| Summary | Risk effective from or to date is less than transaction EFFECTIVE\_FROM\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_016

|  |  |
| --- | --- |
| Summary | Coverage effective from or to date is less than transaction effective from date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_017

|  |  |
| --- | --- |
| Summary | Component effective from or to date is less than transaction effective from date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_018

|  |  |
| --- | --- |
| Summary | TRANSACTION\_APPLIED\_TERM table records that link to TEMP status POLICY\_TERM\_HISTORY table record |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_019

|  |  |
| --- | --- |
| Summary | POLICY\_TERM\_HISTORY table records that have gap in accounting dates |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_020

|  |  |
| --- | --- |
| Summary | Policy with incorrect transaction accounting date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_021

|  |  |
| --- | --- |
| Summary | Agent without row in ENTITY\_ROLE table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_022

|  |  |
| --- | --- |
| Summary | Policy term with EFFECTIVE\_FROM\_DATE greater EFFECTIVE\_TO\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_023

|  |  |
| --- | --- |
| Summary | Policy term that ACCT\_FROM\_DATE greater than ACCT\_TO\_DATE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_024

|  |  |
| --- | --- |
| Summary | Policy premium with rate from date is greater than rate to date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_025

|  |  |
| --- | --- |
| Summary | Policy premium that acct from date is greater than acct to date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_026

|  |  |
| --- | --- |
| Summary | PM trans that Gross premium plus Comp do not equal to Net premium |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_027

|  |  |
| --- | --- |
| Summary | Records with invalid TRANSACTION\_FK in COVERAGE table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_028

|  |  |
| --- | --- |
| Summary | Records with invalid TRANSACTION\_FK in RISK table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_029

|  |  |
| --- | --- |
| Summary | Records with invalid TRANSACTION\_FK in POLICY\_TERM\_HISTORY table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_030

|  |  |
| --- | --- |
| Summary | Records with invalid TRANSACTION\_FK in POLICY\_COVERAGE\_COMPONENT table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: PM\_031

|  |  |
| --- | --- |
| Summary | Records with incorrect TRANSACTION\_FK in POLICY\_COVERAGE\_COMPONENT table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |
| Fix | To be added. |

### TEST ID: PM\_032

|  |  |
| --- | --- |
| Summary | Records with incorrect TRANSACTION\_FK in COVERAGE table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_001

|  |  |
| --- | --- |
| Summary | Invoices that exist in FM\_BILL but not in FM\_BILL\_ACCOUNT table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_002

|  |  |
| --- | --- |
| Summary | Identify FM\_Master records that transaction accounting date is less than minimum accounting date of RISK table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_003

|  |  |
| --- | --- |
| Summary | Identify FM\_Master records that transaction accounting date is less than minimum accounting date of COVERAGE table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_004

|  |  |
| --- | --- |
| Summary | Transaction amount different between FM\_MASTER and FM\_OPEN\_ITEM\_CHANGE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_005

|  |  |
| --- | --- |
| Summary | Transactions with accounting date is unreasonably high |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_006

|  |  |
| --- | --- |
| Summary | Transactions with accounting date is unreasonably low |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_007

|  |  |
| --- | --- |
| Summary | Transactions with effective from date is unreasonably high |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_008

|  |  |
| --- | --- |
| Summary | Transactions with effective from date is unreasonably low |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_009

|  |  |
| --- | --- |
| Summary | Transactions with transaction date is unreasonably high |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_010

|  |  |
| --- | --- |
| Summary | Transactions with transaction date is unreasonably low |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_011

|  |  |
| --- | --- |
| Summary | Accounts that balance do not match between FM\_OPEN\_ITEM and FM\_OPEN\_ITEM\_CHANGE |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_012

|  |  |
| --- | --- |
| Summary | Records in FM\_OPEN\_ITEM table that OPEN\_BALANCE is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: FM\_013

|  |  |
| --- | --- |
| Summary | Records in FM\_OPEN\_ITEM\_CHANGE table that OPEN\_BALANCE is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_001

|  |  |
| --- | --- |
| Summary | CM\_TRANSACTION\_DETAIL w/o matching CIX and claimant on CCX |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_002

|  |  |
| --- | --- |
| Summary | cm\_transaction w/ data\_checking cm\_trans\_status\_history row |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_003

|  |  |
| --- | --- |
| Summary | cm\_transaction w/ missing cm\_trans\_status\_history row |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_004

|  |  |
| --- | --- |
| Summary | cm\_transaction w/o matching CM\_TRANSACTION\_DETAIL row |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_005

|  |  |
| --- | --- |
| Summary | cm\_transaction where trans\_amt <> sum amount on det row |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_006

|  |  |
| --- | --- |
| Summary | txn dtl rows where the txn claim differs from CIX claim |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_007

|  |  |
| --- | --- |
| Summary | entity\_role roles where entity fk is not on entity table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_008

|  |  |
| --- | --- |
| Summary | claimant rows where entity role not on entity role table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_009

|  |  |
| --- | --- |
| Summary | claim\_insured rows where entity role not on entity role table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_010

|  |  |
| --- | --- |
| Summary | CIX rows where claim\_insured\_fk does not exist |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_011

|  |  |
| --- | --- |
| Summary | claims with min claim type and claim history dates are not equal |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_012

|  |  |
| --- | --- |
| Summary | missing cm\_transaction SYS\_UPDATE\_TIME |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_013

|  |  |
| --- | --- |
| Summary | claimant\_fk on txn detail, not on claimant table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_014

|  |  |
| --- | --- |
| Summary | CIX fks on txn detail, not on CIX table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_015

|  |  |
| --- | --- |
| Summary | cm\_trans\_status\_history w/ null status\_start\_date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_016

|  |  |
| --- | --- |
| Summary | cm\_trans\_status\_history rows w/ null accounting\_date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_017

|  |  |
| --- | --- |
| Summary | claims where min txn start date < min claim start date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_018

|  |  |
| --- | --- |
| Summary | CIX rows where POLICY\_FK not on cm\_policy |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_019

|  |  |
| --- | --- |
| Summary | CIX rows where RISK\_FK not on cm\_risk |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_020

|  |  |
| --- | --- |
| Summary | CIX rows where COVG\_FK not on cm\_coverage |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_021

|  |  |
| --- | --- |
| Summary | CIX rows where POLICY\_FK not on policy |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_022

|  |  |
| --- | --- |
| Summary | CIX rows where POLICY\_TERM\_FK not on POLICY\_TERM\_HISTORY table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_023

|  |  |
| --- | --- |
| Summary | CIX rows where RISK\_FK not on RISK table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_024

|  |  |
| --- | --- |
| Summary | CIX rows where COVG\_FK not on COVERAGE table |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_025

|  |  |
| --- | --- |
| Summary | CM\_TRANSACTION\_DETAIL rows where CIX fk is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_026

|  |  |
| --- | --- |
| Summary | CM\_TRANSACTION\_DETAIL rows where claimant fk is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_027

|  |  |
| --- | --- |
| Summary | CM\_TRANSACTION accounting dates not in yyyy/mm/dd format |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_028

|  |  |
| --- | --- |
| Summary | Checking for cm\_trans\_status\_history rows where latest status record does not have the max Trans PK of transaction |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_029

|  |  |
| --- | --- |
| Summary | CM Coverage records with SYS\_UPDATE\_TIME is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_030

|  |  |
| --- | --- |
| Summary | CM Risk records with SYS\_UPDATE\_TIME is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_031

|  |  |
| --- | --- |
| Summary | CM Policy records with s SYS\_UPDATE\_TIME is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_032

|  |  |
| --- | --- |
| Summary | CM Coverage records where EFFECTIVE\_FROM\_DATE is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_033

|  |  |
| --- | --- |
| Summary | CM Risk records where EFFECTIVE\_FROM\_DATE is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_034

|  |  |
| --- | --- |
| Summary | CM Policy records where EFFECTIVE\_FROM\_DATE is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_035

|  |  |
| --- | --- |
| Summary | CM Trans History records where SYS\_CREATE\_TIME is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_036

|  |  |
| --- | --- |
| Summary | CM Trans records whose trans date is less than claim report date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_037

|  |  |
| --- | --- |
| Summary | Invalid claim final disposition date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_038

|  |  |
| --- | --- |
| Summary | Invalid claim type start date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_039

|  |  |
| --- | --- |
| Summary | Invalid claim status start date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_040

|  |  |
| --- | --- |
| Summary | Invalid cm\_policy start date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_041

|  |  |
| --- | --- |
| Summary | Claims that status start date is greater than end date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_042

|  |  |
| --- | --- |
| Summary | Claims that type start date is greater than end date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_043

|  |  |
| --- | --- |
| Summary | Claims that has trans acct date less than policy acct date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_044

|  |  |
| --- | --- |
| Summary | Trans that status start date is greater than status end date |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_045

|  |  |
| --- | --- |
| Summary | Mini policies with overlapping terms |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_046

|  |  |
| --- | --- |
| Summary | Trans with transaction accounting date is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_047

|  |  |
| --- | --- |
| Summary | CIX records that do not have correspond records in CCX |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_048

|  |  |
| --- | --- |
| Summary | Claim transaction detail records where SYS\_UPDATE\_TIME is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_049

|  |  |
| --- | --- |
| Summary | Claim transaction detail records where trans type code is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_050

|  |  |
| --- | --- |
| Summary | Claim transaction detail records where trans status code is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_051

|  |  |
| --- | --- |
| Summary | Claim transaction detail records where trans source fk is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_052

|  |  |
| --- | --- |
| Summary | Claim records that CLAIMANT\_COVERAGE\_XREF link to different claim |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CM\_053

|  |  |
| --- | --- |
| Summary | ENTITY records that CLIENT\_ID is null |
| Description |  |
| Type | Data Inconsistency |
| Fix | To be added. |

### TEST ID: CLAIMS\_CM\_AUTO\_INS\_CLMNTCOV

|  |  |
| --- | --- |
| Parameter | CM\_AUTO\_INS\_CLMNTCOV |
| Description | Indicates if a Claimant/Coverage (a.k.a. feature) should be created automatically when a coverage is added and there is only one claimant on the claim or if a claimant is added and there is only one coverage |
| Type | SYSTEM\_PARAMETER\_UTIL Table Configuration |
| Value | Y or N |

# Future Enhancements

The following enhancements are expected in the future releases:

1. Manage OASIS health check environments through Properties file that will contain the database connection string for environments without passwords.
2. Presently the tool requires the host machine to have a 32-bit oracle client library installed. Ability to work with a 64-bit oracle client library will be addressed.
3. Looping through all the environments works only in file mode. It currently does not work when environments are maintained in the database.