Compliance Tracking for InfoSphere Optim

Terabyte Offerings

August 2013-08-22

V1.1

Optim Terabyte Offerings

The InfoSphere Optim licenses its core offerings using the per Terabyte price metric.

Optim's Core Offerings include:

- Archive (sometimes called Data Growth)
- Test Data Management
- Data Privacy (sometimes called Data Masking)

For these Optim offerings, terabytes are counted as follows:

- Archive is based on the total actual amount of uncompressed data being archived or retired into Optim Archive Files.
- Test Data Management and Data Privacy are based on the overall size of the Data Sources (ie. Production Database tables) that is being managed by the Optim Software.

  When counting terabytes for TDM and Data Privacy, the following should NOT be counted:

  - Free space, space metadata, and other metadata in data pages
  - Indexes, hash access maps, and other high performance access method data
  - Logs
  - Data managed in environments for the sole purpose of validating Optim functionality
  - Data in Disaster Recovery environments (unless used as a direct data source for Optim)
  - Backup copies of the source data (unless used as a direct data source for Optim)
  - Storage used to keep the application code itself
  - Copy storage pools for the space-managed data that reside on removable media.

Definition of Terabyte

A Terabyte is 2 to the 40th power bytes.

Please note: There are two different, commonly used English language definitions for 1TB: two commonly used definitions of 1 TB: 10^{12} and 2^{40} = 1 099 511 627 776 bytes .

For both Terabyte and Gigabyte pricing, Optim uses the "power of two" definition. This definition is explicitly described in the product License Information for each product. Using the wrong definition will translate to a significantly higher price of over 7% for GB pricing and almost 10% for TB pricing.

Optim Process Audit Facility

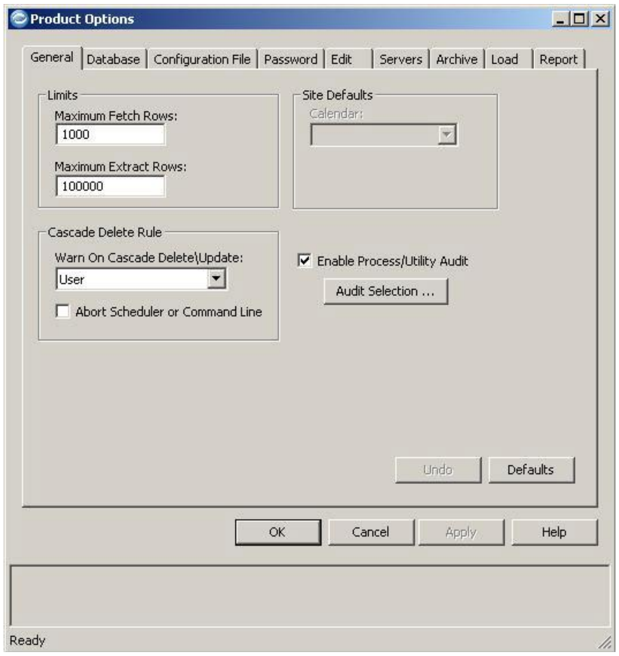
The Optim Process Audit Facility is NOT enabled by default in Optim Distributed V9.1 and below, but for installations tracking usage by TB, it is recommended that the audit facility be enabled. When enabling the audit facility, the Actions and Utilities tabs list the processes for which audit records can be generated. Audit records are generated only for processes that are selected on these tabs. Below are the Actions and Utilities that Optim can audit in Optim V9.1.

Actions	Utilities
• Archive Request • Compare Request • Convert Request • Delete Request • Extract Request • Insert Request • Load Request • Report Request • Restore Request • Table Editor	• Browse • Create • Import • Export • Optim Data Manager (ODM)

Enabling the Optim Process Audit Facility

Instructions for enabling the audit facility can be found in the Optim Installation and Configuration documentation, under Chapter 8, Product Options. The Enable Process/Utility Audit checkbox should be checked and the appropriate Actions and Utilities should be enabled.

For the case of application retirement / decommissioning, Archive Requests need to be enabled to track the amount of data that is archived or retired.



Product Options
General | Database | Configuration File | Password | Edit | Servers | Archive | Load | Report

Limits

Maximum Fetch Rows: 1000

Maximum Extract Rows: 100000

Site Defaults

Calendar:

Cascade Delete Rule

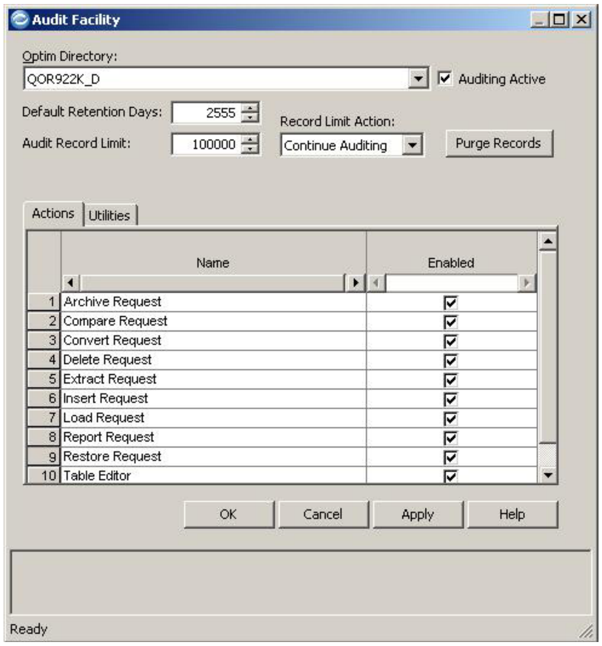
Warn On Cascade Delete\Update: User

☐ Abort Scheduler or Command Line

☑ Enable Process/Utility Audit

Audit Selection ...
Undo
Defaults
OK
Cancel
Apply
Help

Ready



Optim Directory: QOR922K_D

☑ Auditing Active

Default Retention Days: 2555

Audit Record Limit: 100000

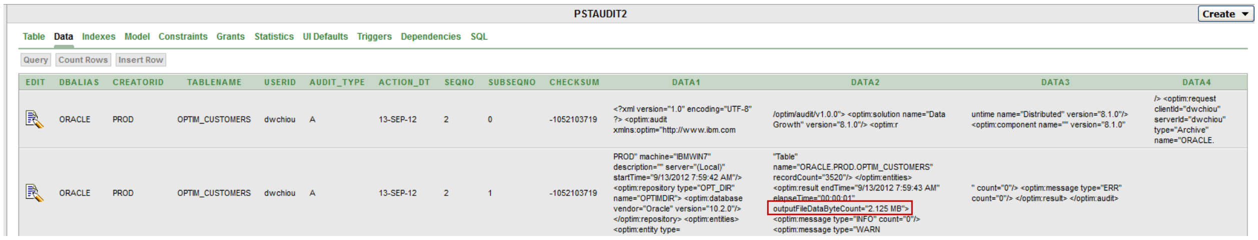
Record Limit Action: Continue Auditing

	Name	Enabled
1	Archive Request	☑
2	Compare Request	☑
3	Convert Request	☑
4	Delete Request	☑
5	Extract Request	☑
6	Insert Request	☑
7	Load Request	☑
8	Report Request	☑
9	Restore Request	☑
10	Table Editor	☑

Ready

Optim Process Audit Records

Process Audit records are stored in the Optim Directory (internal Optim metadata database). The audit table is the PSTAUDIT2 table. The archive file data size is tracked in the XML attribute “outputFileData-ByteCount". This typically appears in the DATA2 column (see below). Audit reports can be periodically run against this table to track TB sizes using any BI reporting tool or simply running a SQL query against the table, like SELECT * FROM *OptimOwnerID*.PSTAUDIT2.

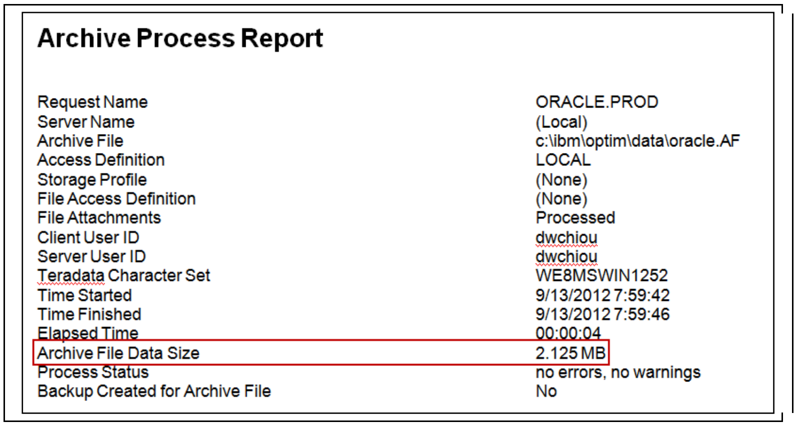


PSTAUDIT2
Create
Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL
Query
Count Rows
Insert Row

EDIT	DBALIAS	CREATORID	TABLENAME	USERID	AUDIT_TYPE	ACTION_DT	SEQNO	SUBSEQNO	CHECKSUM	DATA1	DATA2	DATA3	DATA4
	ORACLE	PROD	OPTIM_CUSTOMERS	dwchiou	A	13-SEP-12	2	0	-1052103719	<?xml version="1.0" encoding="UTF-8" ?> <optim:audit xmlns:optim="http://www.ibm.com	/optim/audit/v1.0.0"> <optim:solution name="Data Growth" version="8.1.0"/> <optim:r	untime name="Distributed" version="8.1.0"/> <optim:component name="" version="8.1.0"	/> <optim:request clientId="dwchiou" serverId="dwchiou" type="Archive" name="ORACLE.
	ORACLE	PROD	OPTIM_CUSTOMERS	dwchiou	A	13-SEP-12	2	1	-1052103719	PROD" machine="IBMWIN7" description="" server="(Local)" startTime="9/13/2012 7:59:42 AM"/> <optim:repository type="OPT_DIR" name="OPTIMDIR"> <optim:database vendor="Oracle" version="10.2.0"/> </optim:repository> <optim:entities> <optim:entity type=	"Table" name="ORACLE.PROD.OPTIM_CUSTOMERS" recordCount="3520"/> </optim:entities> <optim:result endTime="9/13/2012 7:59:43 AM" elapseTime="00:00:01" outputFileDataByteCount="2.125 MB"> <optim:message type="INFO" count="0"/> <optim:message type="WARN	" count="0"/> <optim:message type="ERR" count="0"/> </optim:result> </optim:audit>	

Archive Process Reports

Another option to track usage is through the Archive Process Reports. The Archive Process Report is created after each archive process and reports on the archive file data size (see below). These reports can be saved in a local file directory (user determined location) on the Optim Server. A process to ensure these reports are saved to a common location and not deleted can be established to keep a history of the usage.

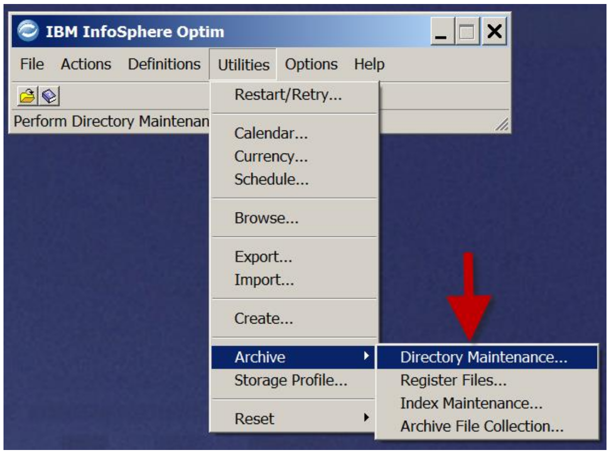


Archive Process Report

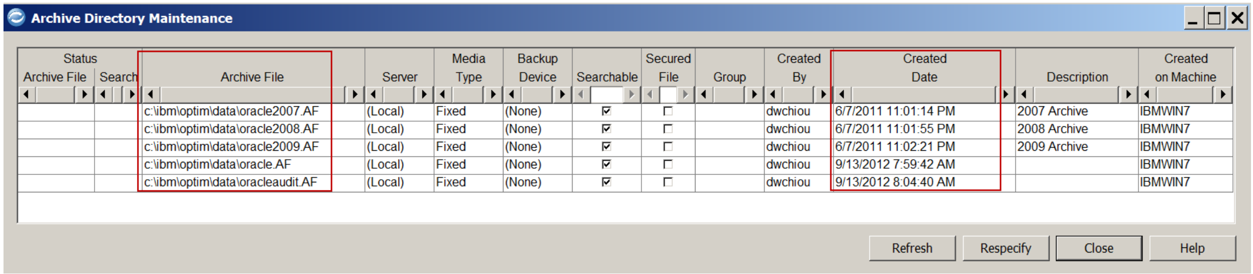
Request Name	ORACLE.PROD
Server Name	(Local)
Archive File	c:\ibm\optim\data\oracle.AF
Access Definition	LOCAL
Storage Profile	(None)
File Access Definition	(None)
File Attachments	Processed
Client User ID	dwchiou
Server User ID	dwchiou
Teradata Character Set	WE8MSWIN1252
Time Started	9/13/2012 7:59:42
Time Finished	9/13/2012 7:59:46
Elapsed Time	00:00:04
Archive File Data Size	2.125 MB
Process Status	no errors, no warnings
Backup Created for Archive File	No

Optim Directory

The Optim Directory automatically maintains a list of all created archive files, so cross-checking the Optim Directory is a way to ensure you have a complete list of archive process reports. A list of all the archive files can be found by selecting the Directory Maintenance option.



The view below provides a list and location of all the archive files and when they were created.



Status Archive File	Search	Archive File	Server	Media Type	Backup Device	Searchable	Secured File	Group	Created By	Created Date	Description	Created on Machine
		c:\ibm\optim\data\oracle2007.AF	(Local)	Fixed	(None)	☑	☐		dwchiou	6/7/2011 11:01:14 PM	2007 Archive	IBMWIN7
		c:\ibm\optim\data\oracle2008.AF	(Local)	Fixed	(None)	☑	☐		dwchiou	6/7/2011 11:01:55 PM	2008 Archive	IBMWIN7
		c:\ibm\optim\data\oracle2009.AF	(Local)	Fixed	(None)	☑	☐		dwchiou	6/7/2011 11:02:21 PM	2009 Archive	IBMWIN7
		c:\ibm\optim\data\oracle.AF	(Local)	Fixed	(None)	☑	☐		dwchiou	9/13/2012 7:59:42 AM		IBMWIN7
		c:\ibm\optim\data\oracleaudit.AF	(Local)	Fixed	(None)	☑	☐		dwchiou	9/13/2012 8:04:40 AM		IBMWIN7

Note: Be aware that looking at the actual size of the archive files will often not match the size indicated in the archive process report. This is because the archive file compresses user data. Even when Optim compression is NOT selected, some simple user data compression may be performed.

Measuring Archive Terabytes for Optim Native z/OS Platform Deployments

Optim native z/OS provides a statistic for the uncompressed volume of user data in an Archive File, in the Optim Archive process report.

These reports should be collected and preserved to document the total volume of data archived.

Here is a sample report. The statistic about the size of the uncompressed user data is highlighted in **bold red** below. It is entitled “Archive File Data Byte Count”:

```
                             ARCHIVE Process Report

Archive File           : USERID.DEMO2.PSTDEMO.ARCV
Access Definition      : USERID.DEMO2.PSTDEMO
Archive Retention Date: None
Archive Expiration     : None
Archive Description    : None
Archive Group          : None
Created by             : Job USERID, using SQLID USERID on DB2 Subsystem DDAF
Time Started           : 2014-12-11 09.12.50
Time Finished          : 2014-12-11 09.12.53

File Compression Impact :
Archive File
  Compression is not available on BASIC or LARGE format datasets.


Process Options:
  Process Mode         : Online
  Archive Data using   : DB2
  Limit Archive Rows   : 3000000

Total Number of Archive Tables               : 5
Total Number of Archived Rows                : 34888
Total Number of First Pass Start Table Rows : 3520
Archive file data byte count : 2,886,106 Bytes (0.003  GB)

     Archived Object Types           Number
     ------------------------------  ------
1    Table-List Tables                    5
2    Related Primary Keys                 5
<Etc.>
```

Don’t Count: Archiving for Test Purposes

Licensees do not need to count terabytes against data that is used to validate Optim’s functionality. For example, if a test environment is set up and test runs are performed, the Test archiving is not counted as consuming Archive TB resource. In an Optim Distributed system, and assuming these test archive activities are sufficiently large as to significantly impact price, Optim records the Test as well as Production archiving, so Test archiving should be identified in a way that allows easy calculation and exclusion of the Test data volume. In an Optim z/OS system they should be excluded from the manual recording and counting.

Measuring TDM or Data Privacy Terabytes for Distributed Platform Deployments

Measuring TDM for Data Privacy Terabytes for a distributed data source depends on the total uncompressed size of the total data sources managed by Optim. Optim supports data sources heterogeneously. That is, a solution can generally process a business object that consists of data from one or more data sources or extract data from one data source or DBMS and insert it in another.

Most of the Optim supported data sources include administrative capabilities that allow you to measure and track the overall storage capacity of the data contained within this data source. Here are some recommendations:

- Relational DBMSes generally gather and can provide the following statistics on a per table or per tablespace basis. These values can be combined by multiplication or division (depending on how the compression ratio is expressed) to get a satisfactory estimate the total uncompressed data in the table:
  - Average row length
  - Row compression ratio
  - Number of rows
- For file data sources, the actual file size can be obtained from the operating system. Adjustments may need to be made if the data is compressed.

See the section below, about “Measuring TDM or Data Privacy Terabytes for Optim native z/OS platform deployments” for further information about measuring z/OS data sources.

Please note that in most cases a data source for TDM and Data Privacy will be a Production Database, but this is not always the case. In some cases, a copy of the Production environment will be created. In these cases, the size of the Production copy user data should be measured.

Measuring TDM or Data Privacy Terabytes for Optim native z/OS platform deployments

Measuring TDM Terabytes for a native z/OS data source depends on the total uncompressed size of the data source.

Note: The z/OS operating system (DFSMS) can compress z/OS files. Optim z/OS makes use of this system compression to compress Archive Files. Users may also compress data sets which are an Optim data source. The LISTCAT utility reports the actual (compressed) data set size and also the uncompressed data set size.

1. For VSAM and Sequential data: use the uncompressed z/OS file size as reported by LISTCAT. This recommendation assumes the VSAM and Sequential data is not compressed programmatically. Such compression is rare in our experience. Optim does provide a user exit to handle compressing / uncompressing VSAM and Sequential data, so if compression is used it must be adjusted for.

2. For DB2 for z/OS: first, see the sample DB2 for z/OS queries below. They have been used at some customer sites to estimate uncompressed DB2 data volume. The discussion here is about the theory.

   DB2 statistics are useful. First, however, you must ensure they are valid and recently run. Then, for any table, simply multiply the average uncompressed row length times the number rows in the table.

   Notes:

   - DB2 has real time statistics as well as user statistics (RUNSTATS). Real time statistics were stored in user defined tables in DB2 for z/OS 8, and in DB2 Catalog Tables since DB2 for z/OS 9. Real time statistics are not always available, specifically for tables defined in older releases of DB2 and where the capability has not been explicitly enabled. Also, they are gathered at the tablespace and indexspace level, and so the information is not necessarily available at the table level. (DB2 is increasingly encouraging / requiring a 1 tablespace per table approach, but tablespaces may still have more than one table in them.) We recommend using the real time statistics if available, since they are gathered automatically and are considered more reliable than RUNSTATS. In the points that follow, however, there are also notes about RUNSTATS statistics.
   - For RUNSTATS statistics: the date of the most recent run of statistics is in column SYSIBM.SYSTABLES.STATSTIME. If statistics have not been run, the STATSTIME value is the minimum permitted value for a timestamp (0001-01-01.00.00.00.000000). This column can be directly updated by an authorized user, so its validity should be checked with the customer DBA.
   - For RUNSTATS statistics: the number of rows in the table is kept in the SYSIBM.SYSTABLES.CARDF column. The value is -1 if statistics have never been run. This column can be directly updated by an authorized user, so its validity should be checked with the customer DBA.
   - If the DB2 tablespace is compressed, then the average row length for a table in the DB2 catalog (available in several tables, but the main one is SYSTABLES.AVGROWLEN) gives the com-

pressed row length. Optim consumption counts the uncompressed volume of data, so the compressed row length by itself is not useful in the Optim consumption calculation.

  - To correct for DB2 tablespace compression, you can determine the compression ratio from a DB2 LOAD or REORG report and use it to compute the uncompressed average row length. For example, if the compression ratio is 80% then the DB2 average row size (compressed) in SYSTABLES.AVGROWLEN must be multiplied by 1/(1-0.80) = 5.
  - Another approach to measuring and correcting for compression is to use the SYSIBM.SYSTABLEPART.PAGESAVE entry. This gives an approximation of the percent savings due to compression. It’s an approximation because both the before compression and after compression values used to compute it include overhead bytes in the DB2 data page. Again, use the formula 1/(1-PAGESAVE/100)) and multiply that by the average (compressed) row length.
- Another option to determine row length is to add up the row length manually based on the individual columns. The average row length of a variable length column can be queried in the DBMS using SQL.

3. For IMS, like DB2, for each segment type, use average segment length times number of segments and sum up across all segment types being managed by TDM.

Sample DB2 for z/OS queries

The following queries have been used at some IBM customer accounts to estimate uncompressed data size:

For non compressed tables

```
SELECT T.NAME, DEC((T.AVGROWLEN*T.CARDF/(1024*1024)),9,2)

FROM SYSIBM.SYSTABLES T

WHERE CREATOR = 'DDS0650' AND

NAME IN ('CUST_ORDER_DETAIL_HIST');
```

For compressed tables

```
SELECT T.NAME,
SUM(DEC(((T.AVGROWLEN*TP.CARD/1048576)*(1/(1-(TP.PAGESAVE*.01)))),9,2))

FROM SYSIBM.SYSTABLES T, SYSIBM.SYSTABLEPART TP

WHERE T.DBNAME = TP.DBNAME

AND T.TSNAME = TP.TSNAME

AND T.CREATOR = 'DDS0650'

AND T.NAME IN ('CUST_ORDER_DETAIL_HIST')

GROUP BY TP.DBNAME,TP.TSNAME,T.NAME;
```