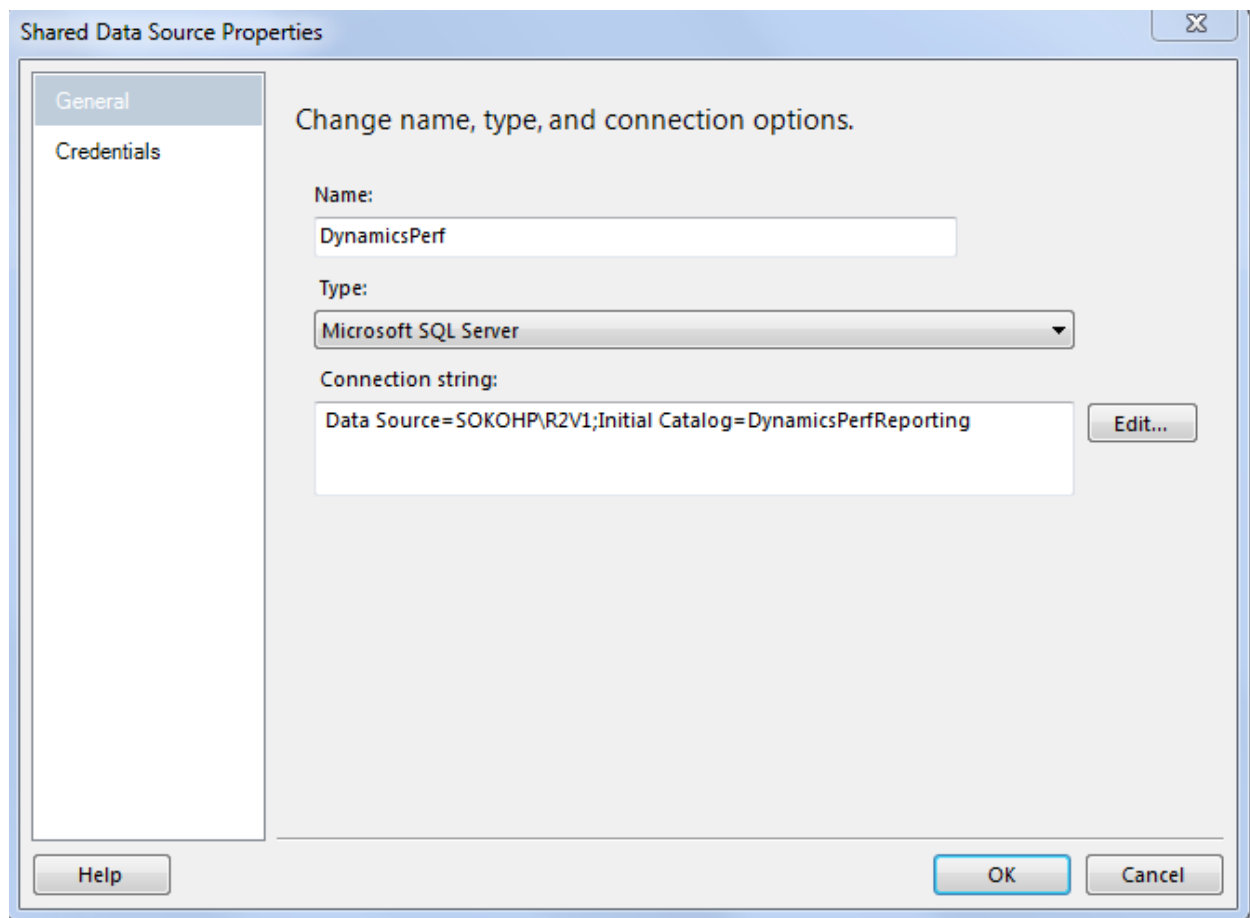


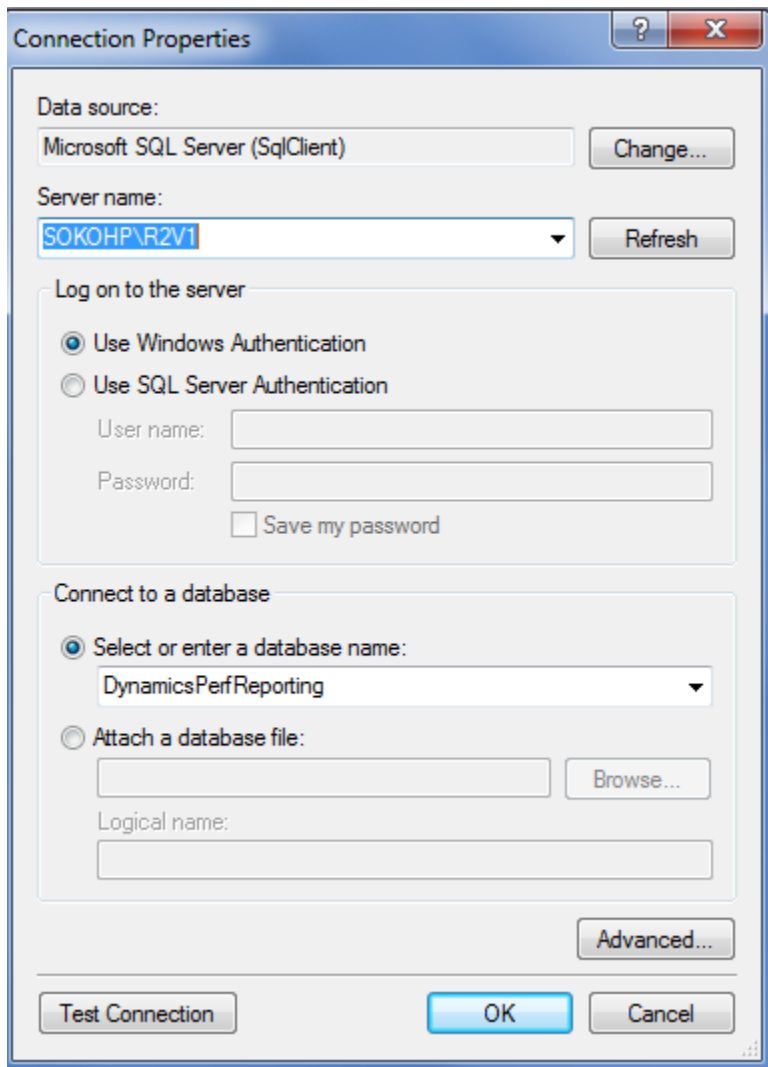
Dynamics Performance Reporting Package

Setup and Deployment:

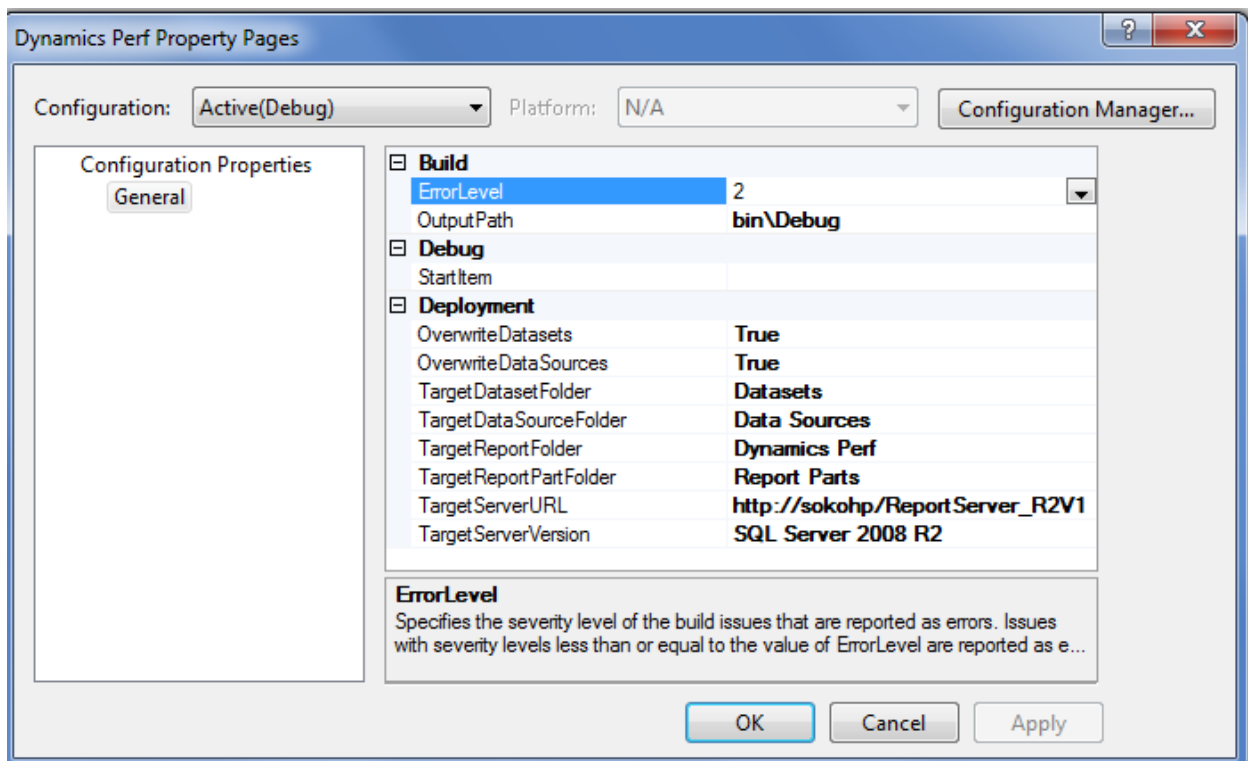
1. Extract the files to any location
2. Open Visual Studio 2008 and select File >> Open >> Project/Solution
3. Point to the Dynamics Perf.sln file and click open
4. Once open go to solution explorer and double click on the DynamicsPerf Shared Data Source



5. Click on Edit and set the server and database to match your environment



6. If you plan on deploying the reports to your report server
 - a. In Solution Explorer right click on the Dynamics Perf Project and go to Properties
7. Change the Report Server to be the report server you want to deploy to



8. To deploy the whole project right click on the Project and select Deploy.
9. If you need to redeploy just one particular report, you can right click on just that report and Deploy.

Configuration Reports:

1. Configuration-Server Configuration

Details: This report pulls server configuration information

Parameters: Database ID

**Page Breaks between each section

Database:

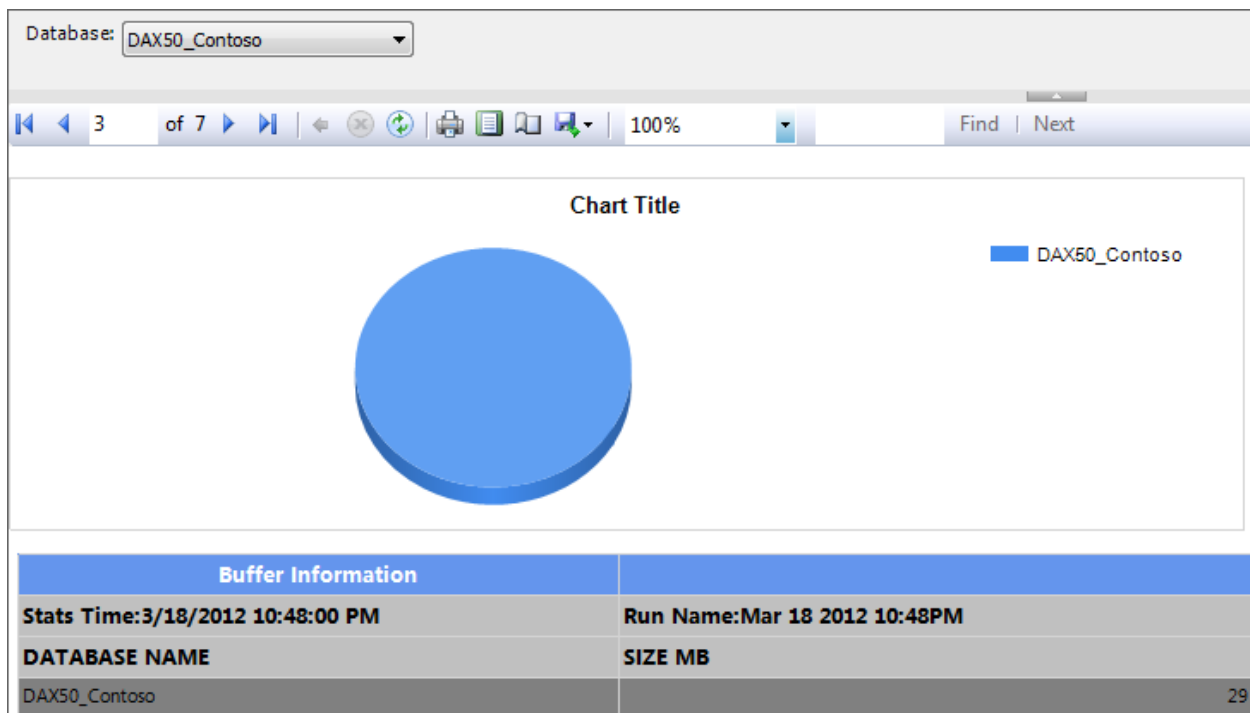
1 of 7 100% Find | Next

Server Information									
Physical Computer Name	Machine Name	Clustered	Character Set	Hyper Ratio	Cpu count	RUN NAME	Bpool Committ ed MB	Bpool Commit Target MB	Page Life Expectan cy
SQL Edition		Product Version	Product Level	Sql Sort Order					
DAX2009L	DAX2009L	0	1	2	2	Feb 26 2012 9:25PM	64	64	6
Developer Edition (64-bit)		10.0.2531.0	SP1	52					
DAX2009L	DAX2009L	0	1	2	2	Mar 18 2012 10:48PM	72	72	22
Developer Edition (64-bit)		10.0.2531.0	SP1	52					

Database:

2 of 7 100% Find | Next

Server Configuration				
Server Start:3/18/2012 10:43:18 PM				
name	minimum	maximum	config value	run value
awe enabled	0	1	0	0
max degree of parallelism	0	64	0	0
max server memory (MB)	16	2147483647	2147483647	2147483647
min server memory (MB)	0	2147483647	0	16
priority boost	0	1	0	0



Database: DAX50_Contoso

4 of 7 100% Find | Next

Virtual Log Files				
DATABASE NAME	FILEID	VLF COUNT	FREE	INUSE

Database Configuration							
Name	ID	compatibility level	recovery model desc	is auto shrink on	compatibility level	is auto create stats on	is auto update stats on
DAX50_Contoso	8	90	SIMPLE	False	90	True	True

Database: DAX50_Contoso

5 of 7 100% Find | Next

Database Files						
Database Name	File Name	Physical Name	File Type	DB Size Mb	DB Free Mb	Growth Units
DAX50_Contoso	AXDB	C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\DAX50_Contoso.mdf	Data	3245	1137	1Mb
DAX50_Contoso	AXDB_log	C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\DAX50_Contoso_1.LDF	Log	379	373	10%

Database: DAX50_Contoso						
<div> <div> 6 of 7 100% Find Next </div> </div>						
SQL Jobs						
job Name	schedule Name	frequency	schedule Time	next Run Date	step name	command
DYNPERF_Capture_Stats	Daily	Daily	17:00 - 23:59	0 / / 00:00	sp_capturestats	EXEC SP_CAPTURESTATS @DATABASE_NAME = 'DAX50_Contoso', @SKIP_STATS='N'
DYNPERF_Capture_Stats_Purge	Daily	Daily	23:59 - 23:59	0 / / 00:00	sp_purgestats	EXEC SP_PURGESTATS @PURGE_DAYS = 14 GO -- Clear Wait Stats DBCC SQLPERF('sys.dm_os_wait_stats', CLEAR);
DYNPERF_PerfStats_Hourly	Hourly	Daily	00:00 - 23:59	0 / / 00:00	Capture Stats	EXEC SP_CAPTURESTATS_PERF @DATABASE_NAME = 'DAX50_Contoso'
DYNPERF_Purge_Blocks	Daily	Daily	04:00 - 23:59	0 / / 00:00	Purge Blocks	[SP_PURGEBLOCKS] @days= 7
SharedServices1_DB_Job_DeleteExpiredSessions	SharedServices1_DB_JobSchedule_DeleteExpiredSessions	Daily	00:00 - 23:59	0 / / 00:00	SharedServices1_DB_JobStep_DeleteExpiredSessions	EXECUTE DeleteExpiredSessions

Documentation Reports:

1. Documentation – AX Long Running Queries

- Details: This report pulls AX Long running Query Information
- Parameters: None. Pulls only records that have a comment.

AX Long Running Queries	
AX Long Duration Query 9	
Date:	10/20/2011 2:30:35 PM
User:	shawn
Company:	
SQL Duration (milliseconds):	14453
SQL Text	
SELECT A.CUSTGROUP,A.SALESTYPE,SUM(B.QTY),B.DIMENSION,SUM(C.LINEAMOUNTMST),SUM(D.LINEAMOUNTMST),E.ITEMGROUPID FROM (oj CUSTINVOICETRANS B LEFT OUTER JOIN CUSTINVOICETRANS C ON ((C.DATAAREAID=?) AND ((C.LINEAMOUNT> ?) AND (B.RECID=C.RECID)))) LEFT OUTER JOIN CUSTINVOICETRANS D ON ((D.DATAAREAID=?) AND ((D.LINEAMOUNT< ?) AND (B.RECID=D.RECID)))) LEFT OUTER JOIN INVENTTABLE E ON ((E.DATAAREAID=?) AND (B.ITEMID=E.ITEMID)),CUSTINVOICEJOUR A WHERE (A.DATAAREAID=?) AND ((B.DATAAREAID=?) AND ((((((((B.INVOICEDATE>=?) AND (B.INVOICEDATE<=?) AND NOT ((B.ITEMID=?))) AND (((((((((((((((B.LEDGERACCOUNT=?) OR (B.LEDGERACCOUNT=?) AND (A.SALESID=B.SALESID)) AND (A.INVOICEID=B.INVOICEID)) AND (A.INVOICEDATE=B.INVOICEDATE)) AND (A.NUMBERSEQUENCEGROUP=B.NUMBERSEQUENCEGROUP))) GROUP BY A.CUSTGROUP,A.SALESTYPE,B.DIMENSION,E.ITEMGROUPID ORDER BY B.DIMENSION,A.CUSTGROUP,E.ITEMGROUPID	
Call Stack	
(S)\Classes\QueryRun\next (S)\Classes\AMC_SalesSRS_TicketSummaryByPeriod\next - line 12	
Comments	
My comment for this query here	

2. Documentation-Long Running Queries

- Details: This report pulls Long running Query Information
- Parameters: None. Pulls only records that have a comment.

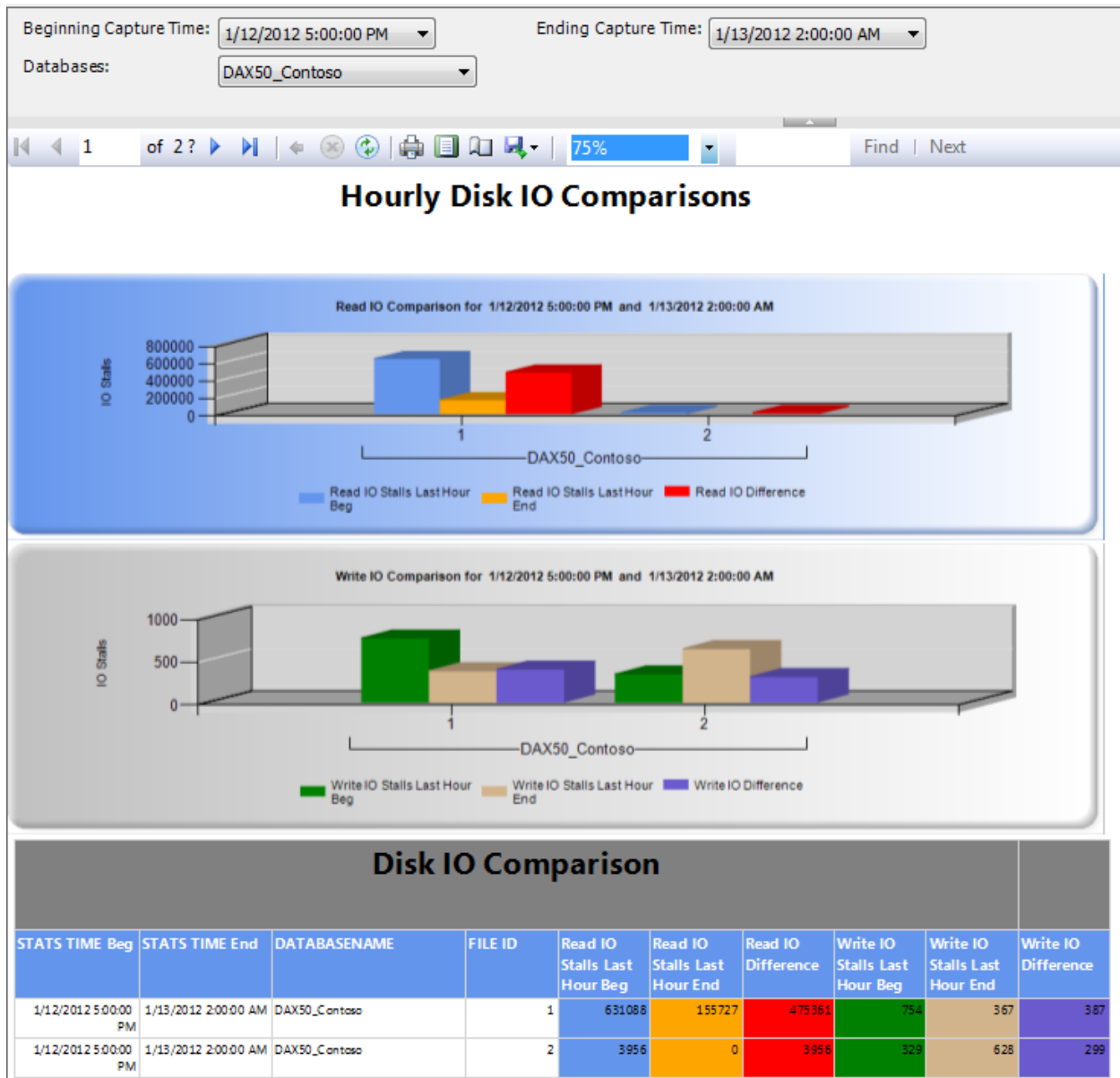
Long Running Query 2	
Total Elapsed Time	691.039
Average Elapsed Time	691.039
Average Logical Reads	1171
2 Execution count	1
SQL Statement	
<pre>delete from AMC_EDILoadSetIds where exists (Select * from SQL2005.RAMAPP.Dbo.XAMDispatchLoad_DivorceLoadSet x where x.LoadSetID = AMC_EDILoadSetIds.LoadSetId) and not exists (select * from SQL2005.RAMAPP.Dbo.xAMDispatchLoad x join SQL2005.RAMAPP.Dbo.xAMLoadSetHeader h on x.loadsetid = h.loadsetid and h.shiptoid = 'DEFAULT' where x.LoadSetID = AMC_EDILoadSetIds.LoadSetId and x.status IN ('D','C')) --PURGE ANY OLD</pre>	
Comments	
My COMMENT Here	

Long Running Query 3	
Total Elapsed Time	104.005
Average Elapsed Time	104.005
Average Logical Reads	1952
3 Execution count	1
SQL Statement	
<pre>Delete From AMC_EDILoadSetIds where not exists (select distinct(AMC_LoadSetID) from PurchTable where PurchTable.AMC_LoadsetID = AMC_EDILoadSetIds.LoadsetID) -- and convert(char(10),AMC_EDILoadSetIds.Crtd_DateTime,102) > = convert(char(10),getdate()-14,102) and convert(char(10),AMC_EDILoadSetIds.Crtd_DateTime,102) > = convert(char(10),getdate()-45,102) --JAS 10/29/2009. Adding logic to populate temp table --Inserting all of the records that will be in the lookup table in AX into our temp table this is to minimize the calls trough the link server</pre>	
Comments	
My COMMENT Here	

Monitoring Reports:

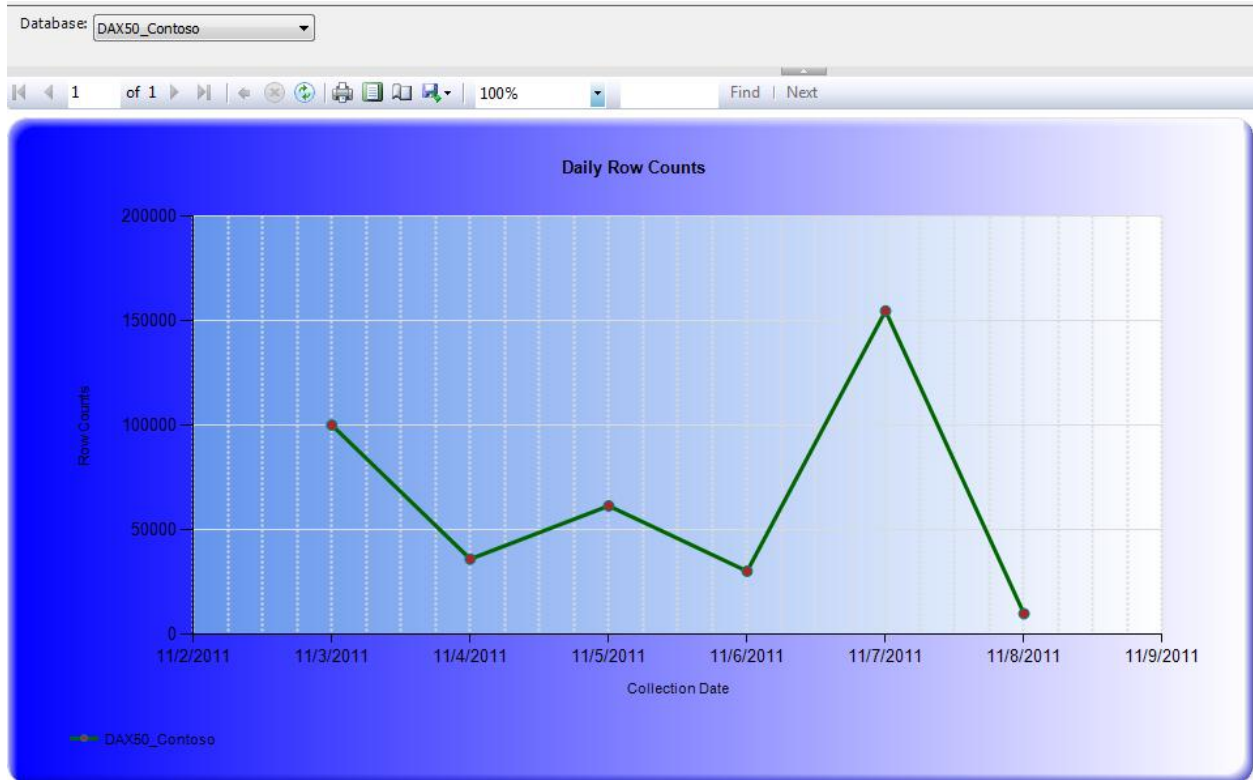
1. Monitoring – Hourly Disk IO

- Details: This report compares Disk IO between two capture periods.
- Parameters: Capture Time Beginning and End and Database



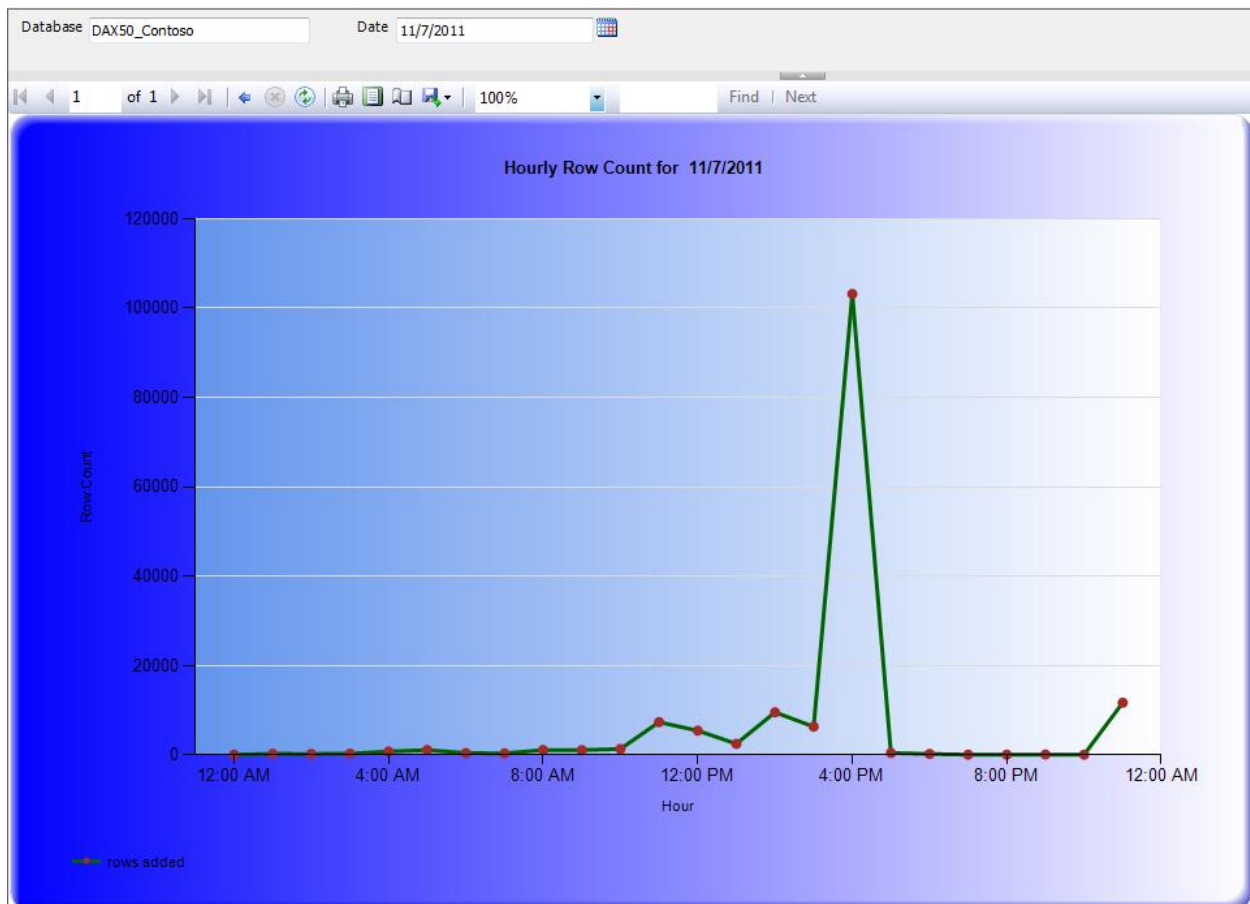
2. Monitoring- Row Count Statistics (Daily).

- a. Details: This report pulls the Daily Row counts
- b. Parameters: Database ID
- c. Actions: Drill thru to Hourly Row Counts Report



3. Monitoring-Row Count Statistics Hourly.

- a. Details: This report pulls the Hourly Row counts
- b. Parameters: Database ID and Date
- c. Actions:
 - Drills into Hourly Row Count Details Report
 - This is a sub report or the Row Count Statistics (Daily) Report to show details of a data point in that report



4. Monitoring- Row Count Statistics (Hourly Details)

- a. Details: This report pulls the Hourly Row counts Details
- b. Parameters: Database ID and Statstime
- c. Actions:
 - This is a sub report or the Hourly Row Counts Report to show details of a data point in that report

Statstime 11/7/2011 4:00:00 PM Database DAX50_Contoso

1 of 2 ? 100% Find | Next

Hourly Row Count Details

STATS TIME	DATABASE NAME	ROWRANK	TABLE NAME	ROWS ADDED
11/7/2011 4:00:00 PM	DAX50_Contoso	1	INVENTTRANSPosting	14456
11/7/2011 4:00:00 PM	DAX50_Contoso	2	LEDGERTRANS	10138
11/7/2011 4:00:00 PM	DAX50_Contoso	3	INVENTTRANS	7276
11/7/2011 4:00:00 PM	DAX50_Contoso	4	INVENTREPORTDIMHISTORY	7247
11/7/2011 4:00:00 PM	DAX50_Contoso	5	SALESPARMLINE	7146
11/7/2011 4:00:00 PM	DAX50_Contoso	6	CUSTINVOICETRANS	7095
11/7/2011 4:00:00 PM	DAX50_Contoso	6	SALESLINE	7095
11/7/2011 4:00:00 PM	DAX50_Contoso	7	LEDGERBALANCESDIMTRANS	4618
11/7/2011 4:00:00 PM	DAX50_Contoso	7	LEDGERBALANCESTRANS	4618
11/7/2011 4:00:00 PM	DAX50_Contoso	8	SMMTRANSLOG	4375
11/7/2011 4:00:00 PM	DAX50_Contoso	9	AMC_TKTDET_PROGS	2642
11/7/2011 4:00:00 PM	DAX50_Contoso	10	TRANSACTIONLOG	2344
11/7/2011 4:00:00 PM	DAX50_Contoso	11	CUSTTRANS	2311
11/7/2011 4:00:00 PM	DAX50_Contoso	11	CUSTTRANSIDREF	2311
11/7/2011 4:00:00 PM	DAX50_Contoso	12	SALESPARMSUBTABLE	2216
11/7/2011 4:00:00 PM	DAX50_Contoso	12	SALESPARMTABLE	2216
11/7/2011 4:00:00 PM	DAX50_Contoso	12	SALESPARMUPDATE	2216
11/7/2011 4:00:00 PM	DAX50_Contoso	13	CUSTINVOICEJOUR	2166
11/7/2011 4:00:00 PM	DAX50_Contoso	13	CUSTINVOICESALESLINK	2166
11/7/2011 4:00:00 PM	DAX50_Contoso	13	SALESTABLE	2166
11/7/2011 4:00:00 PM	DAX50_Contoso	14	CUSTTRANSOPEN	1608

5. Monitoring – RowCountComparison

- Details: This report compares row counts for two capture times.
- Parameters: Beginning and End Run Times
- Actions:
- Can drill into Monitoring-RowCountComparisonSub from either pie chart

Beginning Run Time: Jan 12 2012 5:00PM Ending Run Time: Jan 13 2012 5:00PM

1 of 1 Whole Page Find Next

Record Count by MB and Rows

Comparing: Jan 12 2012 5:00PM and Jan 13 2012 5:00PM

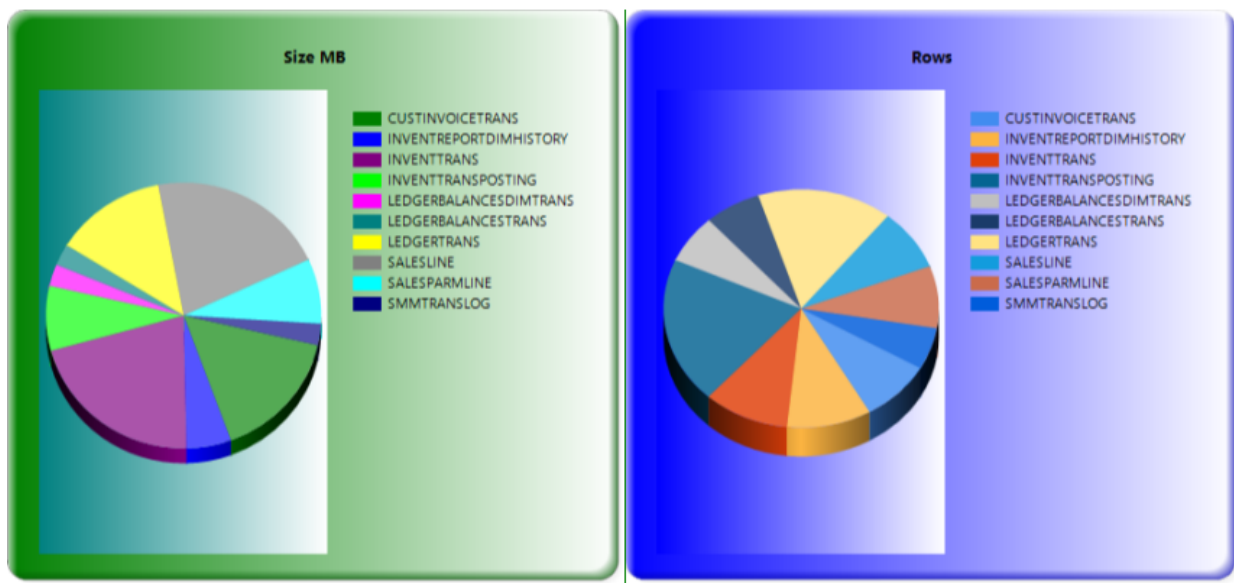


TABLE NAME	Original Page Count	New Page Count	Original SIZE MB	New SIZE MB	Delta SIZE MB	Table Updates	Delta In Rows	Days
INVENTTRANSPosting	181901	182329	1421	1424	3	16246	16246	1
LEDGERTRANS	329591	330148	2574	2579	5	28505	12961	1
INVENTTRANS	299222	300226	2337	2345	8	32956	8231	1
INVENTREPORTDIMHISTORY	68426	68673	534	536	2	10327	8096	1
SALESPARMLINE	151003	151339	1179	1182	3	9174	6887	1
CUSTINVOICETRANS	342017	342800	2672	2678	6	13665	6841	1
SALESLINE	299645	300657	2340	2348	8	20869	6824	1
LEDGERBALANCESDIMTRANS	46420	46517	362	363	1	2684	5417	1
LEDGERBALANCESTRANS	40958	41037	319	320	1	2684	5405	1
SMMTRANSLOG	51318	51425	400	401	1	6169	4634	1

6. Monitoring-RowCountComparisonSub

- Details: This report shows row count details of a certain table
- Parameters: Beginning and End Run Times and table

Beginning Run Time	Jan 12 2012 5:00PM	<input type="checkbox"/> NULL	Ending Run Time	Jan 13 2012 5:00PM	<input type="checkbox"/> NULL
Table	INVENTTRANS	<input type="checkbox"/> NULL			

1 of 1

100%

Find | Next

Row Count Comparison Details

TABLE NAME	Original Page Count	New Page Count	Original SIZEMB	New SIZEMB	Delta SIZEMB	Table Updates	Delta In Rows	Days
INVENTTRANS	299222	300226	2337	2345	8	32956	8231	1

7. Monitoring-Wait Statistics

- Details: This report shows wait statistics compared between two capture periods
- Parameters: Beginning and End Run Times

Beginning Stat Time: 1/12/2012 5:00:00 PM

Ending Stat Time: 1/13/2012 12:00:00 AM

1 of 1100%Find | Next

Wait Statistics

WAIT TYPE	STATS TIME BEG	STATS TIME END	Waiting Tasks Last Hour Beg	Waiting Tasks Last Hour End	Wait Time ms Last Hour Beg	Wait Time ms Last Hour End	RANK Beg	Rank END
ASYNC_NETWORK_IO	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	10165	9513	26333	3630	12	23
BACKUPIO	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	1777	1458270	1800	3107674	20	10
BROKER_TO_FLUSH	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	1758	1757	1799304	1799308	9	12
CHECKPOINT_QUEUE	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	11	11	3595164	3595998	7	7
FT_ITS_SCHEDULER_IDLE_WAIT	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	60	59	3600026	3540025	5	8
LAZYWRITER_SLEEP	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	16401	21208	7192580	7188242	1	2
LCK_M_X	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	499	477	703	990	23	31
LOGMGR_QUEUE	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	18450	10779	3598997	3598707	6	6
MSQL_XP	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	3776	3757	35503	37647	10	16
OLEDB	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	32399	4214908	3922	171701	16	14
PAGEIOLATCH_EX	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	80	2022	583	10758	25	18
PAGEIOLATCH_SH	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	2214	19027	13926	68994	14	15
PAGELATCH_EX	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	50241	27717	1088	1083	22	30
PREEMPTIVE_OLEDBOPS	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	135985	96497	3644	3329	17	24
PREEMPTIVE_OS_AUTHENTICATIONOPS	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	13280	13362	4677	4392	15	22
PREEMPTIVE_OS_GETPROCADDRESS	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	3776	3757	35421	37577	11	17
PREEMPTIVE_OS_LOOKUPACCOUNTSID	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	4508	4358	2318	2251	18	26
REQUEST_FOR_DEADLOCK_SEARCH	1/12/2012 5:00:00 PM	1/13/2012 12:00:00 AM	720	720	3600217	3600229	3	3

Query Analysis Reports:

1. Query Analysis (Level 1) Table Activity

- Details: This report shows read and write activity per table.
- Parameters: None
- Actions: Can drill into Query Analysis (Level 2) Index Density from a table

Table Activity (Level One)

DATABASE NAME	TABLE NAME	Ratio Of Reads	Ratio Of Writes	Total Read Operations	Total Write Operations	Total Operations
DAX50_Contoso	SYSCONFIG	0.22689	0.77311	81	276	357
	BATCH	0.56522	0.43478	221	170	391
	AIFRUNTIMECACHE	0.50000	0.50000	138	138	276
	BATCHJOB	0.68571	0.31429	96	44	140
	SYSCIENTSESSIONS	0.67949	0.32051	53	25	78
	BATCHGLOBAL	0.69388	0.30612	34	15	49
	SYSSERVERSESSIONS	0.64103	0.35897	25	14	39
	BATCHJOBHISTORY	0.60000	0.40000	12	8	20
	BATCHCONSTRAINTS	0.88000	0.12000	44	6	50
	SYSLASTVALUE	0.87234	0.12766	41	6	47
	EPSTATESTORE	0.25000	0.75000	2	6	8
	SYSBREAKPOINTS	0.75000	0.25000	15	5	20
	SYSEVENT	0.33333	0.66667	2	4	6
	SYSBREAKPOINTLIST	0.85714	0.14286	24	4	28
	SYSTEMSEQUENCES	0.66667	0.33333	6	3	9
	EPSEVERSTATECLEANUPSETTINGS	0.75000	0.25000	6	2	8
	WORKFLOWMESSAGETABLE	0.66667	0.33333	4	2	6
	SYSSERVERCONFIG	0.90000	0.10000	9	1	10
	SQLSYSTEMVARIABLES	0.98214	0.01786	55	1	56
	BATCHHISTORY	1.00000	0.00000	6	0	6
	BATCHJOBALERTS	1.00000	0.00000	2	0	2
	SYSCLUSTERCONFIG	1.00000	0.00000	1	0	1
	BATCHSERVERCONFIG	1.00000	0.00000	1	0	1
	BATCHSERVERGROUP	1.00000	0.00000	79	0	79
	ACCESSRIGHTSLIST	1.00000	0.00000	64	0	64
	SYSUSERINFO	1.00000	0.00000	2	0	2
	SQLDICTIONARY	1.00000	0.00000	208	0	208
	SYSSETUPLOG	1.00000	0.00000	280	0	280
	SYSSIGNATURESETUP	1.00000	0.00000	2	0	2

2. Query Analysis (Level 2) Index Density

- Details: This report shows index density
- Parameters: None
- Actions: Can drill into Query Analysis (Level 3) Index Histogram from an index

Table

Database

1 of 1

Find | Next

100%

Index Density (Level Two)

DATABASENAME	TABLERNAME	INDEXNAME	DENSITY	LEN	COLUMNS
DAX50_Contoso	SQLDICTIONARY	_WA_Sys_00000001_7F60E D59	0.00038	4	TABLEID
DAX50_Contoso	SQLDICTIONARY	_WA_Sys_00000002_7F60E D59	0.00193	4	FIELDID
DAX50_Contoso	SQLDICTIONARY	_WA_Sys_00000004_7F60E D59	0.00009	23	NAME
DAX50_Contoso	SQLDICTIONARY	_WA_Sys_00000005_7F60E D59	0.00007	23	SQLNAME
DAX50_Contoso	SQLDICTIONARY	_WA_Sys_00000007_7F60E D59	0.00483	4	STRSIZE
DAX50_Contoso	SQLDICTIONARY	_WA_Sys_00000008_7F60E D59	0.50000	4	SHADOW
DAX50_Contoso	SQLDICTIONARY	_WA_Sys_0000000B_7F60E D59	0.50000	4	FLAGS
DAX50_Contoso	SQLDICTIONARY	I_65518FIELD	0.00038	4	TABLEID
DAX50_Contoso	SQLDICTIONARY	I_65518FIELD	0.00002	8	TABLEID, FIELDID
DAX50_Contoso	SQLDICTIONARY	I_65518FIELD	0.00002	12	TABLEID, FIELDID, ARRAY
DAX50_Contoso	SQLDICTIONARY	I_65518FIELD	0.00002	16	TABLEID, FIELDID, ARRAY, SHADOW
DAX50_Contoso	SQLDICTIONARY	I_65518RECID	0.07143	8	RECID
DAX50_Contoso	SQLDICTIONARY	I_65518RECID	0.07143	12	RECID, TABLEID
DAX50_Contoso	SQLDICTIONARY	I_65518RECID	0.07143	16	RECID, TABLEID, FIELDID
DAX50_Contoso	SQLDICTIONARY	I_65518RECID	0.07143	20	RECID, TABLEID, FIELDID, ARRAY
DAX50_Contoso	SQLDICTIONARY	I_65518RECID	0.07143	24	RECID, TABLEID, FIELDID, ARRAY, SHADOW

3. Query Analysis (Level 3) Index Histogram

- Details: This report shows index histogram information
- Parameters: None
- Actions: Can drill into Query Analysis (Level 4) Queries Calling Index from an index

Table	SQLDICTIONARY	Column	TABLEID					
Database	DAX50_Contoso							
<div>5 of 6 ? 100% Find Next</div>								
DATABASE NAME	TABLE NAME	INDEX NAME	COLUMN NAME	RANGE HI KEY	RANGE ROWS	EQ ROWS	DISTINCT RANGE ROWS	AVG RANGE ROWS
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8476	123	99	8	15
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8492	210	7	14	14
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8502	72	149	9	8
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8522	218	65	16	13
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8544	258	11	21	12
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8584	261	16	30	8
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8612	224	22	19	11
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8636	217	8	12	17
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	8698	262	40	24	11
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	12150	252	32	18	13
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	16008	136	63	5	26
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20008	202	65	16	12
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20023	148	99	13	11
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20051	203	77	12	16
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20127	165	32	14	11
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20149	163	43	13	12
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20167	257	8	12	20
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20211	257	14	24	10
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20256	258	23	21	12
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	20288	250	27	24	10
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	65498	260	8	14	18
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	65529	259	11	21	12
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	65533	64	8	3	21
DAX50_Contoso	SQLDICTIONARY	WA Sys 00000001 7F60ED59	TABLEID	65534	0	5	0	1
DAX50_Contoso	SQLDICTIONARY	I 65518FIELD	TABLEID	1	0	39	0	1
DAX50_Contoso	SQLDICTIONARY	I 65518FIELD	TABLEID	13	257	41	9	27
DAX50_Contoso	SQLDICTIONARY	I 65518FIELD	TABLEID	20	88	49	6	14
DAX50_Contoso	SQLDICTIONARY	I 65518FIELD	TABLEID	41	195	101	16	11
DAX50_Contoso	SQLDICTIONARY	I 65518FIELD	TABLEID	51	149	49	9	16

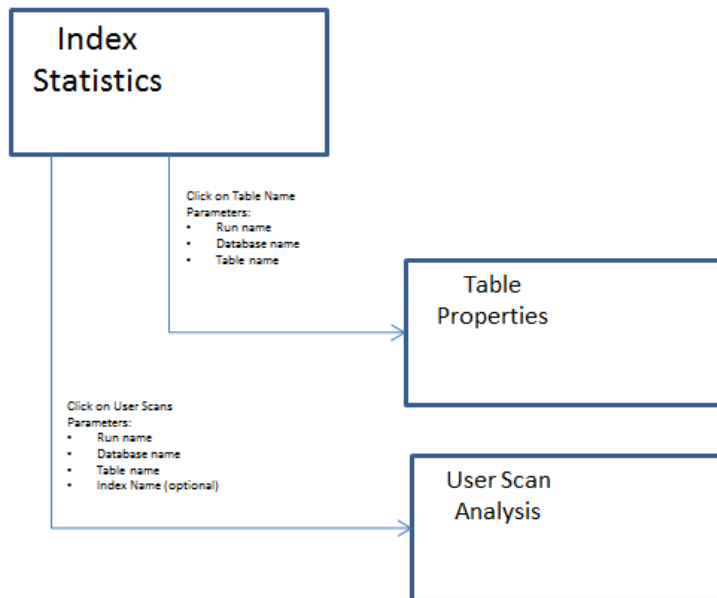
4. Query Analysis (Level 4) Queries Calling Index

- Details: This report shows the queries that call an index from step 3
- Parameters: None
- Notes: This is currently not formatted to print on a standard size sheet, but it does allow you to see all columns in the view

Index I_65518FIELD											
1 of 1 100% Find Next											
RUN NAME	DATABASE NAME	ROW NUM	QUERY HASH	EXECUTION COUNT	TOTAL ELAPSED TIME	AVG ELAPSED TIME	AVG LOGICAL READS	AVG ROWS RETURNED	SQL TEXT	QUERY PARAMETER VALUES	QUERY PLAN
Mar 18 2012 10:48PMs	DAX50_Contoso	22824	#Error	69	388.022	5.623	316	0	SELECT A.TABLEID.A.RECID FROM SQLDICTIONARY A WHERE (((FLAGS=@P1) AND (SHADOW=@P2)) AND (STRSIZE IN (@P3)))	 	<ShowPlanXML xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan" Version="1.1" Build="10.0.2766.0"><BatchSequence><Statements><StmtCursor or StatementText="(@P1 int.@P2 int.@P3 int) SELECT A.TABLEID.A.RECID FROM SQLDICTIONARY A WHERE (((FLAGS=@P1) AND (SHADOW=@P2)) AND (STRSIZE IN (@P3)))" StatementId="1" StatementComplid="1" StatementType=" DECLARE CURSOR" ><Cursor Plan CursorName="" CursorActualType="FastForward" CursorRequestedType="FastForward" CursorConcurrency="Read Only" ForwardOnly="true"><Operation

Scan Analysis Reports:

Scenario: Scan Analysis



1. Scan Analysis – Index Statistics

- Details: This report shows the index statistics for a set of tables in a database
- Parameters: Run Time, Database, Tables, Indexes and Record Count
- Action: Can drill into either Scan Analysis – Table Properties Report from Table or Scan Analysis – User Scan Analysis Report from User Scans Value

Design

Preview

Run name

Mar 18 2012 10:48PM

Databases:

DAX50_Contoso

Tables:

BATCH, BATCHCONSTRAINTS,

Indexes:

I_124NAME, I_2096COMPANY

Record Count:

100

1 of 1

100%

Find | Next

Index Statistics

Database					
Table					
INDEX NAME	INDEX DESCRIPTION	INDEX KEYS	USER SEEKS	USER SCANS	USER UPDATES
DAX50_Contoso					
BATCH					
I_2827BATCHIDX	NONCLUSTERED	BATCHJOBID, STATUS, MODIFIEDDATETIME	4	0	56
I_2827BATCHJOBID	NONCLUSTERED	BATCHJOBID, RUNTIMETASK	4	0	2
I_2827RECID	CLUSTERED, UNIQUE, PRIMARY KEY	RECID	82	131	56
I_2827STATUSUSER	NONCLUSTERED	GROUPID, STATUS, RUNTYPE, CREATEDBY, PRIVATETASK, COMPANY	0	0	56
BATCHCONSTRAINTS					
I_2100BATCHID	CLUSTERED	BATCHID, DEPENDSONBATCHID, EXPECTEDSTATUS	2	0	2
I_2100RECID	NONCLUSTERED, UNIQUE, PRIMARY KEY	RECID	0	2	2
I_2100SERVERSEARCH	NONCLUSTERED	BATCHID, EXPECTEDSTATUS, DEPENDSONBATCHID	26	14	2
BATCHGLOBAL					
I_124NAME	CLUSTERED, UNIQUE, PRIMARY KEY	NAME	34	0	15
BATCHHISTORY					
I_2272BATCH	NONCLUSTERED	BATCHID			

2. Scan Analysis – Table Properties

- Details: This report shows the table properties from a table selected in previous report
- Parameters: Run Time, Database, Tables

Run Time: Databases:

Tables:

1 of 1 100% Find | Next

Table Properties

DATABASE NAME	TABLE NAME	OCC ENABLED	CACHE LOOKUP	TABLE GROUP
DAX50_Contoso	BATCH	True	0	3

3. Scan Analysis – User Scan Analysis

- Details: This report shows the user scan analysis from value selected in Index Statistics Report
- Parameters: Run Time, Database, Tables

Runtime: Database:

Table Name: Number of Records:

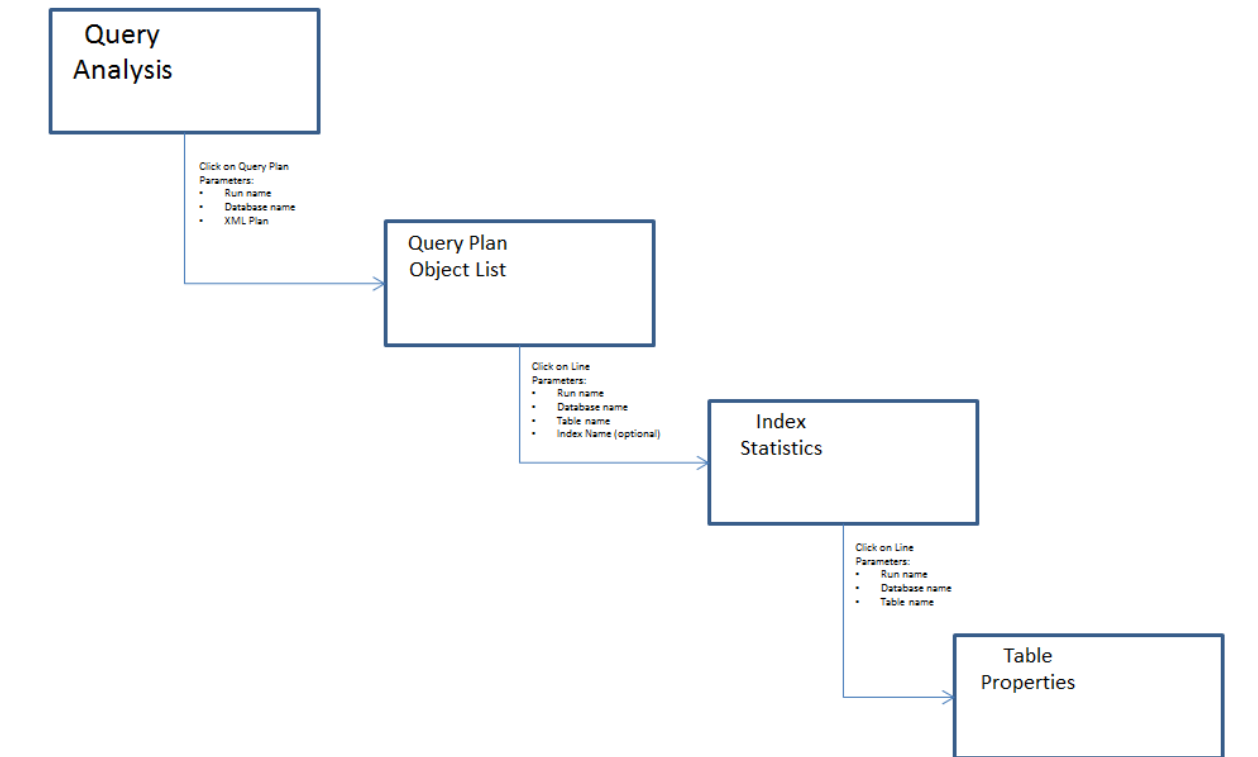
1 of 2 ? 100% Find | Next

User Scan Analysis

EXECUTION COUNT	TOTAL ELAPSED TIME	AVG ELAPSED TIME	AVG LOGICAL READS	PHYSICAL OPERATOR	TABLE NAME	INDEX NAME
Run Name: Mar 18 2012 10:48PM						
Database: DAX50_Contoso						
4	6.00	1.50	20	Index Scan	CUSTTABLE	I_077OURACCOUNTNUMIDX
5	5.00	1.00	20	Index Scan	CUSTTABLE	I_077OURACCOUNTNUMIDX
9	11.00	1.22	21	Index Scan	CUSTTABLE	I_077OURACCOUNTNUMIDX
4	5.00	1.25	20	Index Scan	CUSTTABLE	I_077OURACCOUNTNUMIDX
6	8.00	1.33	22	Index Scan	CUSTTABLE	I_077OURACCOUNTNUMIDX

General Query Analysis Reports:

Scenario: General Query Analysis



1. General Query Analysis-Query Analysis

- Details: This report shows the Query Analysis Details
- Parameters: Run Time, Database, Number of Records

Runtime:	Mar 18 2012 10:48PM	Database	DAX50_Contoso
Number of Records:	100		
1 of 2 ?			
100%			
Find Next			
<h3>Query Analysis</h3>			
Run Name: Mar 18 2012 10:48PM		Database: DAX50_Contoso	
EXECUTION COUNT	TOTAL ELAPSED TIME	AVG ELAPSED TIME	AVG LOGICAL READS
Query Hash			
17	806557.132	47444.537	5983037
0x0748D61474899D44			
17	806557.132	47444.537	5983037
0x0748D61474899D44			

2. General Query Analysis-Query Plan Object List

**In Progress

Query Analysis Misc. Reports:

1. Query Analysis Misc. – Expensive Queries

- Details: This report list most expensive queries by Total Elapsed Time
- Parameters: Database

Database DAX50_Contoso						
<div> <div>1 of 2 ?</div> <div>100%</div> <div>Find Next</div> </div>						
Expensive Queries						
DATABASE NAME	EXECUTION COUNT	QUERY HASH	TOTAL ELAPSED TIME	AVG ELAPSED TIME	TOTAL LOGICAL READS	AVG LOGICAL READS
DAX50_Contoso	1352040	#Error	11327300.896	8.377	365686297	270
SELECT PrimaryEventID FROM xAMC_ARStatementLog WHERE EventType = @EventType AND PrimaryEventID = @PrimaryEventID AND CONVERT(char(10), ProcessDate, 102) = CONVERT(char(10), GETDATE(), 102) AND Consolidated = @Consolidated						
DAX50_Contoso	1273780	#Error	10620266.465	8.337	343191838	269
SELECT PrimaryEventID FROM xAMC_ARStatementLog WHERE EventType = @EventType AND PrimaryEventID = @PrimaryEventID AND CONVERT(char(10), ProcessDate, 102) = CONVERT(char(10), GETDATE(), 102) AND Consolidated = @Consolidated						
DAX50_Contoso	53	#Error	4853310.593	91571.897	263420237	4970193
INSERT INTO @ResultTable exec xAMC_RPT_AR_CustAging @DataAreaId, @AgingDate, @TransDate, @Name, @GroupByInvoiceAccount						
DAX50_Contoso	49	#Error	4473603.875	91298.038	242320719	4945320

2. Query Analysis Misc. – SuggestedIndexes

- Details: This report list Indexes that are suggested by SQL
- Parameters: Tables

Tables: [ACCESSRIGHTSLIST], [ADDRESS]									
1 of 2 ? 100% Find Next									
Suggested Indexes									
SQL TEXT	INDEX IMPACT	TABLE NAME	EQUALITY COLUMNS	INEQUALITY COLUMNS	INCLUDED COLUMNS	EXISTENTIAL	TRANSITIVE	AVG	TOTAL LOGICAL READS
UPDATE @MasterTable SET CustList = COALESCE(CustList + ';' + @AccountNum, @AccountNum) WHERE ID = @MasterID	33.2412	[CUSTTABLE]		[AMC_MASTERCUSTID]		9,830,441.00	1,956,459.84	0.20	9.00 88655433
UPDATE @MasterTable SET CustList = COALESCE(CustList + ';' + @AccountNum, @AccountNum) WHERE ID = @MasterID	33.2412	[CUSTTABLE]		[AMC_MASTERCUSTID]		9,812,861.00	1,952,914.64	0.20	9.00 88493126
UPDATE @MasterTable SET CustList = COALESCE(CustList + ';' + @AccountNum, @AccountNum) WHERE ID = @MasterID	33.2412	[CUSTTABLE]		[AMC_MASTERCUSTID]		9,730,821.00	1,936,563.71	0.20	9.00 87722457