

# SQL Server 2005 Analysis Services (SSAS) Server Properties

SQL Server Technical Article

Writers: Anthony T. Mann, Edward Melomed

Published: June 2006

Applies To: SQL Server 2005

### Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

 2006 Microsoft Corporation. All rights reserved.

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

### Table of Contents

[Executive Summary 1](#_Toc136412543)

[About This Paper 1](#_Toc136412544)

[Setting Server Properties 1](#_Toc136412545)

[Analysis Server Properties Page 1](#_Toc136412546)

[Surface Area Configuration Manager 3](#_Toc136412547)

[Introduction 3](#_Toc136412548)

[Instances 3](#_Toc136412549)

[Languages 5](#_Toc136412550)

[Memory 5](#_Toc136412551)

[Process Memory 5](#_Toc136412552)

[Jobs 6](#_Toc136412553)

[Thread Pools 6](#_Toc136412554)

[How This Document Is Organized 7](#_Toc136412555)

[Server Properties 8](#_Toc136412556)

[AggregationMemoryLimitMax 8](#_Toc136412557)

[AggregationMemoryLimitMin 8](#_Toc136412558)

[AllowAdHocOpenRowsetQueries 9](#_Toc136412559)

[AllowedBrowsingFolders 10](#_Toc136412560)

[AllowedProvidersInOpenRowset 10](#_Toc136412561)

[AllowSessionMiningModels 11](#_Toc136412562)

[BackupDir 11](#_Toc136412563)

[BufferMemoryLimit 12](#_Toc136412564)

[BuiltinAdminsAreServerAdmins 13](#_Toc136412565)

[CheckDistinctRecordSortOrder 13](#_Toc136412566)

[CreateQueryLogTable 14](#_Toc136412567)

[CollationName 15](#_Toc136412568)

[CommitTimeout 15](#_Toc136412569)

[ComUdfEnabled 16](#_Toc136412570)

[CoordinatorCancelCount 16](#_Toc136412571)

[CoordinatorExecutionMode 17](#_Toc136412572)

[DatabaseConnectionPoolMax 18](#_Toc136412573)

[DatabaseConnectionPoolConnectTimeout 18](#_Toc136412574)

[DataDir 19](#_Toc136412575)

[DeepCompressValue 20](#_Toc136412576)

[DefaultDrillthroughMaxRows 20](#_Toc136412577)

[DSO - LocksDirectory 21](#_Toc136412578)

[DSO - RemoteLocksDirectory 22](#_Toc136412579)

[DSO - RemoteRepositoryConnectionString 22](#_Toc136412580)

[DSO - RepositoryConnectionString 23](#_Toc136412581)

[EnableBinaryXML 23](#_Toc136412582)

[EnableCompression 24](#_Toc136412583)

[ExternalCommandTimeout 25](#_Toc136412584)

[ExternalConnectionTimeout 25](#_Toc136412585)

[FlightRecorder - Enabled 26](#_Toc136412586)

[FlightRecorder - FileSizeMB 27](#_Toc136412587)

[FlightRecorder - LogDurationSec 27](#_Toc136412588)

[FlightRecorder - SnapshotDefinitionFile 28](#_Toc136412589)

[FlightRecorder - SnapshotFrequencySec 29](#_Toc136412590)

[FlightRecorder - TraceDefinitionFile 29](#_Toc136412591)

[ForceCommitTimeout 30](#_Toc136412592)

[IdleConnectionTimeout 30](#_Toc136412593)

[IdleOrphanSessionTimeout 31](#_Toc136412594)

[InstanceVisible 31](#_Toc136412595)

[IPV4Support 32](#_Toc136412596)

[IPV6Support 33](#_Toc136412597)

[Language 33](#_Toc136412598)

[LazyProcessing - Enabled 34](#_Toc136412599)

[LazyProcessing - MaxObjectsInParallel 35](#_Toc136412600)

[LazyProcessing - SleepIntervalSecs 35](#_Toc136412601)

[LinkFromOtherInstanceEnabled 36](#_Toc136412602)

[LinkInsideInstanceEnabled 36](#_Toc136412603)

[LinkToOtherInstanceEnabled 37](#_Toc136412604)

[ListenOnlyOnLocalConnections 37](#_Toc136412605)

[LogDir 38](#_Toc136412606)

[LowMemoryLimit 38](#_Toc136412607)

[MaxConcurrentPredictionQueries 39](#_Toc136412608)

[MaxIdleSessionTimeout 40](#_Toc136412609)

[MaxThreads 40](#_Toc136412610)

[MemoryLimitErrorEnabled 41](#_Toc136412611)

[MinIdleSessionTimeout 41](#_Toc136412612)

[MinThreads 42](#_Toc136412613)

[Port 43](#_Toc136412614)

[QueryLogConnectionString 43](#_Toc136412615)

[QueryLogSampling 44](#_Toc136412616)

[QueryLogTableName 45](#_Toc136412617)

[RequireClientAuthentication 45](#_Toc136412618)

[ROLAPDimensionProcessingEffort 46](#_Toc136412619)

[ServerTimeout 47](#_Toc136412620)

[ServiceAccountIsServerAdmin 47](#_Toc136412621)

[TempDir 48](#_Toc136412622)

[TotalMemoryLimit 48](#_Toc136412623)

[Conclusion 49](#_Toc136412624)

#### Executive Summary

SQL Server 2005 Analysis Services (SSAS) enables you to fine tune and control the behavior of SSAS. It does this through the use of properties that are available for configuration at the “instance-level.” These properties are known as server properties.

The behaviors that are controlled by server properties affect how SSAS processes partitions, cubes, and dimensions. They also affect performance, security, and the resources required by SSAS.

This whitepaper discusses the server properties that can be changed by a database administrator or system administrator and are listed in alphabetical order. It also gives some high-level information about the way SSAS manages memory, process memory, instances, languages, and thread pools.

#### About This Paper

##### Setting Server Properties

All server property settings are stored in a text file in XML format, named msmdsrv.ini. This file is located, by default, in the <*drive*>:\Program Files\Microsoft SQL Server\<*Instance Folder*>\OLAP\Config folder. You can edit these settings using any XML text editor, including Notepad or Visual Studio. However, the easiest way to change these settings is in the Graphical User Interface (GUI) tools provided with SQL Server 2005. Each of the server properties presented in this white paper can be changed in one of two places (depending on the property) in the SQL Server 2005 GUI:

• Analysis Server Properties page

• Surface Area Configuration (SAC) Manager

**Note** All screenshots shown in this white paper use the TCP/IP address 192.168.17.16. To configure Analysis Services properties on your own computer, use your TCP/IP address or server name.

**Note** The following conventions are used throughout this white paper:

<*drive*> - The hard drive or volume where you installed SQL Server 2005.

<*Instance Folder*> - The folder under which SQL Server 2005 components have been installed. For example, this location might be *MSSQL.1*, *MSSQL.2*, or *MSSQL.3*. This folder depends on the order you install SQL Server 2005 components and the number of components you install.

###### Analysis Server Properties Page

Analysis Server enables you to change some of the server properties through a properties page, which lists numerous properties listed on the page. To open this page, follow these steps:

1. Open SQL Server Management Studio.

2. Ensure that you open a connection to Analysis Server (not Database Engine).

3. In the Object Explorer pane, right-click the server to which you have connected.

4. Click the Properties menu item.

5. Ensure the General page is selected, as it will be by default.

6. Click the Show Advanced (All) Properties check box. You’ll see the screen shown in Figure 1.



Figure 1: Analysis Server Properties (General) Page

7. You will see the following fields listed (columns) for each property (rows):

• **Name.** The name of the server property.

• **Value.** This is the field that you change if you want the value updated.

• **Current Value.** The value currently used by Analysis Server.

• **Default Value.** The value used if no value is specified.

• **Restart.** Indicates whether a restart of the SQL Server Analysis Services service is required before your newly entered Current Value will take effect. The name of the service depends on the instance name. For the default instance, the service is named MSSQLServerOLAPService. For a named instance, it follows the format of MSOLAP$<InstanceName>.

• **Units.** Shows the datatype of the property.

• **Category.** Indicates whether this is a Basic or Advanced server property. If you uncheck the Show Advanced (All) Properties check box, all server properties in the Advanced category will be hidden.

**Note** A value of False in any server property on Analysis Server Properties page will result in a value of 0 stored in the msmdsrv.ini file. Conversely, a value of True in Analysis Server Properties page will result in a value of 1 stored in the Msmdsrv.ini file.

8. If you wish to change any server property, enter your desired value in the Current Value field. You can change as many values as you want.

9. Click OK to save your changes.

###### Surface Area Configuration Manager

In addition to property pages, the SAC Manager can be used to change some of the properties mentioned in this white paper. Open the SAC Manager by clicking its icon in the Configuration Tools group under SQL Server 2005 on the Start menu. This brings up a dialog box to prompt you to select one of the two areas of configuration, as shown in Figure 2.

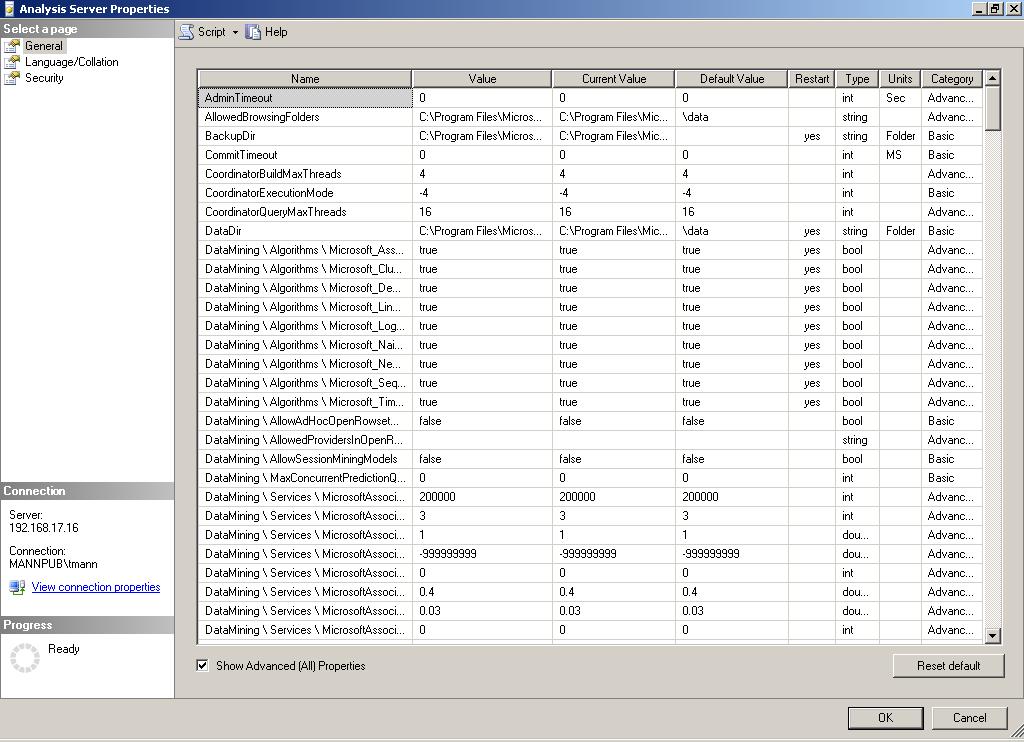


Figure 2: Surface Area Configuration Options

#### Introduction

This white paper does not aim to teach the concepts of SQL Server 2005 Analysis Services, but a few points are important to understand so that you can make informed decisions about when to change property values.

##### Instances

A client or other server can connect to an instance of SSAS in one of two ways. It can connect to the instance directly if it knows the port under which it is listening. By default, the SSAS default instance listens on TCP port 2383 and named instances of SSAS are assigned a port dynamically. When an instance starts, it registers itself with the SQL Server Browser service. An application can query this service on port 2382, which will redirect the connection to the correct SSAS instance. However, the application still has to know the name of the instances to get the port number from SQL Server Browser service. It will not enumerate all installed instances. Note that the port number 2382 for SSAS in the SQL Server Browser service is different than the port for the SQL Server Database engine, which listens on port 1434.

If the SQL Server Browser service is not running, a client must connect directly to the named instance by using its port number. Most of the client applications will accept “ServerName:PortNum” format for the server name when connecting. The tricky part is that the named instance ports are dynamic, so they may not receive the same port number the next time the instance is started. See Figure 3 to see how the instances are configured.

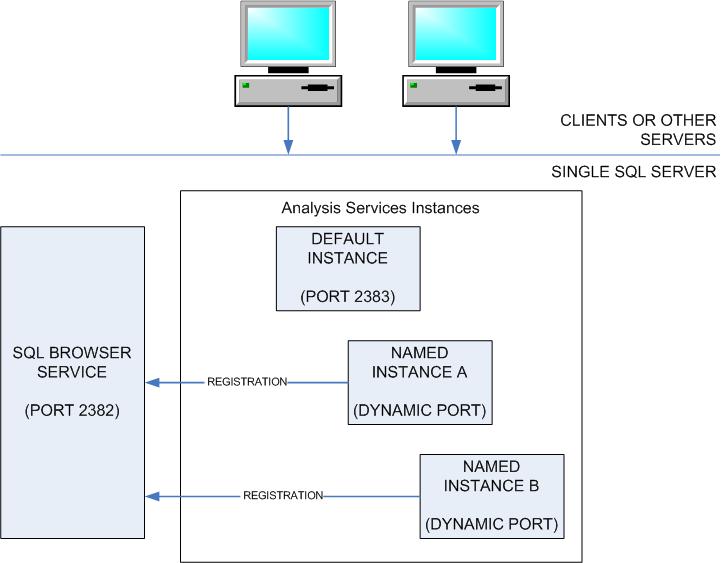


Figure 3: Analysis Services Instances Default Behavior

The SQL Server Browser service is shared between the SQL Server 2005 Database engine and the Analysis Services engine. For the Database engine, for security purposes, it is recommended to turn the SQL Server Browser service off. This can create a dilemma because you may need it for the Analysis Services engine. It is a recommended practice to turn the service on if you need it, but otherwise leave it off.

Alternatively, you can configure which port each instance listens on. For more information on doing this, see the **Port** server property.

##### Languages

The default language used in your SSAS installation is determined by the language of the product inside the box. In other words, the language is tied to the SQL Server 2005 SKU that you purchased.

SSAS can return dimension members in a language requested by the client application, however the **Language** property specifies the default language returned to the client. This property also controls which language is used to create databases. The language setting is preserved inside the database, so it is backed up and restored in the selected language. For example, suppose ServerA is set to English and ServerB is set to French. A database that is backed up from ServerA to ServerB would have its language set to English. Therefore, after the restore to ServerB, you would likely want to change the database language to French.

##### Memory

SSAS has a special memory “cleaner” background thread that constantly determines if it needs to clean up memory. The cleaner looks at the amount of memory used. The following basic processes are used by the cleaner to control amount of physical memory used by Analysis Server:

• If the memory used is above the value set in the **TotalMemoryLimit** property, the cleaner cleans up to this value.

• If the memory used is under the value set in the **LowMemoryLimit** property, the cleaner does nothing.

• If the memory used is between the values set in the **LowMemoryLimit** and the **TotalMemoryLimit** properties, the cleaner cleans memory on the need-to-use basis.

If the value specified in any of these properties is between 0 and 100, the value is treated by SSAS as a percentage of total physical memory. If the value specified is greater than 100, the value is treated by SSAS as an absolute memory value (in bytes). Note that when Analysis Server is processing, if it requires additional memory above the value specified in **TotalMemoryLimit**, it will try to reserve that amount, regardless of the **TotalMemoryLimit** value.

##### Process Memory

Process memory is an area of memory that is reserved for processing dimensions, cubes, and partitions in SSAS. If you are going to change any of the memory settings under the Process node, it is important to understand how the processing works before doing so.

**Note** Processing a partition or dimension could result in more that one job created, in which case memory is calculated per job, and not per processing command.

For an SSAS job, Analysis Server calculates the memory required for processing and asks the memory governor (the global component keeping track of memory usage by Analysis Server) for that amount of memory. Using **BufferMemoryLimit**, you can limit the amount of memory available for a processing job, which allows more processing jobs to run in parallel. Setting this property to a smaller value could cause Analysis Server to start swapping data to disk, which would effectively slow down the server.

If the governor cannot give at least the minimum amount of memory required, an error will be returned (assuming the **MemoryLimitErrorEnabled** server property value is **True**).

##### Jobs

During processing and querying of the Dimensions or partitions Analysis Server for some operations will start multiple internal job opearations. For example sending command to process a single dimension will will result in processing multiple attributes in parallel. Each attribute processing handled by a single unit of execution — a job. The number of jobs running in parallel is controlled by the **CoordinatorExecutionMode** server property. During different stages of job execution, Analysis Server might start several threads. The number of threads is controlled through the thread pool mechanism.

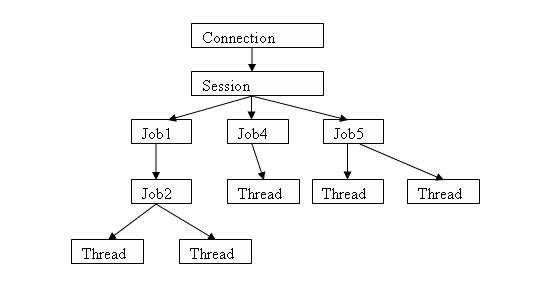


Figure 4: Relationship of Jobs and Threads

##### Thread Pools

When a query is issued, a thread for the query is spawned by the Formula Engine component. The Formula Engine requests data from the Storage Engine component by passing sub-cube requests. The Storage Engine gets data from the disk and also populates a data cache to be used to answer similar queries in the future. The Storage Engine asks for data from the storage system on multiple threads.

The **MinThreads** and **MaxThreads** server properties (for Process and Query) control the behavior of the threads in a thread pool.

**Note** One thread pool exists for queries and one exists for the processing pool. The Process pool is used by the Storage Engine for query execution as well.

#### How This Document Is Organized

To help you understand each server property listed in this white paper, these attributes are described:

|  |  |
| --- | --- |
| Property Name | Tells which property page and corresponding property name is being described. The Analysis Server Properties page is shown in Figure 1.  Note: All server property changes are stored in the msmdsrv.ini file. |
| Default Value | Lists the value that will be used if no explicit value is set or if the msmdsrv.ini file is missing or deleted. |
| Unit of Measure | Shows the unit associated with the Default Value, Minimum Value, or Maximum Value. |
| Data Type | Shows the Analysis Server 2005 datatype that is used in setting a server property value. |
| Minimum Value | Indicates the lowest possible value that you can specify for the given server property. |
| Maximum Value | Indicates the highest possible value that you can specify for the given server property. |
| Requires Restart | Indicates whether the SQL Server Analysis Services service needs to be restarted for a new value to go into effect. The name of the service depends on the name of your Analysis Services instance. This value is shown as either Yes or No. |
| Alternate GUI Tool | Shows whether a given property can be configured with the SAC tool in addition to the property page. If the property can be configured with the SAC tool, it will be listed as one of the following:  SAC/Services – Use the Surface Area Configuration for Services and Connections tool (see Figure 2). All properties are listed under the Analysis Services node.  SAC/Features – Use the Surface Area Configuration for Features tool (see Figure 2). All properties are listed under the Analysis Services node. |
| Special Notes | Describes special situations, circumstances, or ramifications that you need to keep in mind before you set the property value. |
| Security Implications | Shows how the security of your system may be affected by changing the given server property. |

#### Server Properties

##### AggregationMemoryLimitMax

**AggregationMemoryLimitMax** is a server property that controls the maximum amount of memory that can be used for SSAS aggregation processing. This value is expressed as a percentage of total memory if the value is less than 100, or an absolute value of bytes if the value is greater than 100. The default value is 80, which indicates that a maximum of 80% of the total physical memory can be used for SSAS aggregation processing.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Process \ AggregationMemoryLimitMax |
| Default Value | 80 |
| Unit of Measure | Percentage if between 0 and 100.  Bytes if greater than 100. |
| Data Type | Double |
| Minimum Value | 0 – SSAS will build aggregations for a single partition segment at a time, swapping rest of the segments to disc. |
| Maximum Value | For Percentage: 100 – SSAS will use up to 100% of total physical memory.  For Bytes: 1.79E + 308 is the maximum value that can be specified. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### AggregationMemoryLimitMin

**AggregationMemoryLimitMin** is a server property that controls the minimum amount of memory that will be used for SSAS aggregation processing. This value is expressed as a percentage of total memory if the value is less than 100, or an absolute value of bytes if the value is greater than 100. The default value is 10, which indicates that a minimum of 10% of the total physical memory will be used for SSAS aggregation processing.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Process \ AggregationMemoryLimitMin |
| Default Value | 10 |
| Unit of Measure | Percentage if between 0 and 100.  Bytes if greater than 100. |
| Data Type | Double |
| Minimum Value | 0 – SSAS will not guarantee that memory gets allocated for Aggregation processing. |
| Maximum Value | For Percentage: 100 – SSAS will use up to 100% of total physical memory.  For Bytes: 1.79E + 308 is the maximum value that can be specified. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### AllowAdHocOpenRowsetQueries

**AllowAdHocOpenRowsetQueries** is a server property that specifies if an Analysis Services query will allow the use of the OPENROWSET Transact-SQL statement in a user.

|  |  |
| --- | --- |
| Property Name | General Page: Data Mining \ AllowAdHocOpenRowsetQueries |
| Default Value | False – AdHoc OpenRowset queries are not allowed. |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – AdHoc OpenRowset queries are not allowed. |
| Maximum Value | True – AdHoc OpenRowset queries are allowed. |
| Requires Restart | No |
| Alternate GUI Tool | SAC/Features: Ad-Hoc Remote Queries \ Enable OPENROWSET and OPENDATASOURCE support |
| Special Notes | None |
| Security Implications | Using OPENROWSET enables a client to connect to an OLE DB data source, so the call to OPENROWSET needs to specify the provider name, as well as login information. If your application does not need this functionality, it is recommended to set the value to False. |

##### AllowedBrowsingFolders

**AllowedBrowsingFolders** is a server property that specifies a list of folders (directories) that can be browsed when saving files, opening files, and browsing files in Analysis Services Backup, Restore, ParittionProperties, and Synchronization dialog boxes.

|  |  |
| --- | --- |
| Property Name | General Page: AllowedBrowsingFolders |
| Default Value | Two folders are browsed by default:  *<drive>*:\Program Files\Microsoft SQL Server\*<Instance Folder>*\OLAP\Log | *<drive>*:\Program Files\Microsoft SQL Server\*<Instance Folder>*\OLAP\Backup |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Multiple folders need to be delimited by the pipe (|) symbol.  Analysis Services does not check the validity of the path you enter for this server property. Ensure the path exists when changing this property.  If you would like to make a whole drive available for browsing just change this property to *<drive>*:\ |
| Security Implications | It is possible to specify a folder that contains sensitive or improper data. Make sure you specify folders that will contain applicable Analysis Services data. |

##### AllowedProvidersInOpenRowset

**AllowedProvidersInOpenRowset** is a server property that specifies which OLE DB providers (ProgIDs) are allowed in an OPENROWSET query.

|  |  |
| --- | --- |
| Property Name | General Page: DataMining \ AllowedProvidersInOpenRowset |
| Default Value | No default value is specified. |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Multiple folders need to be delimited by a semicolon (;) symbol.  If a value of [All] is specified, all OLE DB providers registered on the server can be used in OPENROWSET queries.  If the value is blank, no OLE DB providers can be used in OPENROWSET queries. |
| Security Implications | It is possible to specify an OLE DB provider that is unsafe or causes performance problems. Specify only OLE DB providers that you know to be safe, such as those provided by Microsoft. |

##### AllowSessionMiningModels

**AllowSessionMiningModels** is a Boolean server property that dictates whether mining models are enabled at the session level.

|  |  |
| --- | --- |
| Property Name | General Page: DataMining \ AllowSessionMiningModels |
| Default Value | False |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Session Mining Models are not allowed. |
| Maximum Value | True – Session Mining Models are allowed. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### BackupDir

**BackupDir** is a server property that specifies where Analysis Server backups are stored by default.

|  |  |
| --- | --- |
| Property Name | General Page: BackupDir |
| Default Value | Determined during setup, but is here by default:  *<drive>*:\Program Files\Microsoft SQL Server\*<Install Folder>*\OLAP\Backup |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | A backup operation that explicitly specifies a backup path will ignore this property.  This folder is specifically reserved for Analysis Server. It is not the same folder used for the Database Engine.  Analysis Services does not check the validity of the path you enter for this server property. Ensure the path exists when changing this property. |
| Security Implications | If you change this property, you may need to create the folder and also set the appropriate security permissions on it. |

##### BufferMemoryLimit

**BufferMemoryLimit** is a server property that controls the amount of memory that is used to store/cache data coming from relational data sources and for processing cubes, dimensions and partitions. This value is expressed as a percentage of total memory if the value is less than 100, or an absolute value of bytes if the value is greater than 100. The default value is 60, which indicates that a maximum of 60 percent of the total physical memory can be used as the buffer memory limit. See the Process Memory Section earlier in this paper for a detailed discussion of how the **BufferMemoryLimit** server property works. It is used in conjunction with other process server properties.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Process \ BufferMemoryLimit |
| Default Value | 60 |
| Unit of Measure | Percentage if between 0 and 100.  Bytes if greater than 100. |
| Data Type | Double |
| Minimum Value | 0 – SSAS will limit memory available for processing to the absolute minimum. |
| Maximum Value | For Percentage: 100 – SSAS will use up to 100% of total physical memory for buffer memory.  For Bytes: 1.79E + 308 is the maximum value that can be specified. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### BuiltinAdminsAreServerAdmins

BuiltinAdminsAreServerAdmins is a Boolean server property that dictates whether the built-in (local) Administrators group is to be treated as an Analysis Services administrator.

|  |  |
| --- | --- |
| Property Name | General Page: Security \ BuiltinAdminsAreServerAdmins |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Local Administrators cannot perform tasks as a member of the Analysis Services Administrators group. |
| Maximum Value | True – Local Administrators can perform tasks as a member of the Analysis Services Administrators group. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Please note this property is not guiarantee protection against local Administarators. Local machine administrators have wide variety of security privileges.This property is not to be used as protection mechanism against local Administrators. |
| Security Implications | While this property is set to True by default, you should consider whether there are any security implications in your specific environment. One example would be a case where a server administrator manages the database engine and other server-related tasks (DHCP, DNS, etc.), but you want to maintain tight control over who administers Analysis Services. |

##### CheckDistinctRecordSortOrder

**CheckDistinctRecordSortOrder** is a Boolean server property that dictates how record sort orders will be handled. When processing distinct counts coming from a relational data source, Analysis Services needs to have records in a specific order to ensure accurate counting.

If collations are different between Analysis Services and the relational source, an error might occur if this property is set to **True**. On the other hand, if collations are known to be the same, performance can be enhanced by setting the property to **False**, but no checking will occur.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Process \ CheckDistinctRecordSortOrder |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – No order checking will occur between a relational data source and Analysis Services. This effectively turns off error checking, but increases the risk that counting of distinct records will be inaccurate. |
| Maximum Value | True – Records order is checked when data is coming from a relational data source into Analysis Services. With this value set to True, errors will be generated if order is not the same. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### CreateQueryLogTable

**CreateQueryLogTable** is a Boolean server property that dictates whether Analysis Server will attempt to create a table in a relational database to collect query statistics. The name of the table is specified by the **QueryLogTableName** server property. For more information about the SSAS query log, see the article “[Configuring the Analysis Services Query Log](http://go.microsoft.com/fwlink/?linkid=66164)”.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ QueryLog \ CreateQueryLogTable |
| Default Value | False |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### CollationName

**CollationName** is a server property that indicates which collation Analysis Services should use.

|  |  |
| --- | --- |
| Property Name | Language/Collation Page: Windows Collation |
| Default Value | Specified during the setup of Analysis Services. |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | It is important to know that SSAS uses only Windows collations, not SQL Server collations. |
| Security Implications | None |

##### CommitTimeout

Analysis Server processing operations need to acquire a write lock before it can commit a transaction. In order to acquire a write lock, no other read locks can be taken by another process or query. Therefore, Analysis Services needs to wait until all read locks are released. The transaction will wait for a period of time to acquire a write lock, as specified by the **CommitTimeout** property before rolling back.

|  |  |
| --- | --- |
| Property Name | General Page: CommitTimeout |
| Default Value | 0 |
| Unit of Measure | Milliseconds |
| Data Type | Integer |
| Minimum Value | 0 – Indicates that Analysis Services will wait indefinitely to acquire a write lock in order to commit a transaction. |
| Maximum Value | 2147483647 Milliseconds, or approximately 25 days. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### ComUdfEnabled

**ComUdfEnabled** is a Boolean server property that indicates whether COM-based User-Defined functions are to be enabled in your Analysis Server. This property is available for legacy purposes, as SQL Server 2005 enables the use of CLR User-Defined functions.

|  |  |
| --- | --- |
| Property Name | General Page: Feature \ ComUdfEnabled |
| Default Value | False |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Indicates that COM User-Defined functions are not enabled. |
| Maximum Value | True – Indicates that COM User-Defined functions are enabled. |
| Requires Restart | No |
| Alternate GUI Tool | SAC/Features: User-Defined Functions \ Enable loading of User-Defined COM functions |
| Special Notes | None |
| Security Implications | COM objects cannot be secured as well as CLR assemblies, so enable this feature only if it is required. |

##### CoordinatorCancelCount

**CoordinatorCancelCount** is a server property that manages how queries and processing operations are canceled. Internally, as a query is executing, SSAS iterates through execution code, called the *coordinator*. For maximum responsiveness to cancel queries, the SSAS coordinator can check whether the user canceled the operation once per iteration. However, this frequency can actually hurt performance. Therefore, by default, the coordinator checks whether the user canceled a query or processing only once for every 1000 internal iterations.

|  |  |
| --- | --- |
| Property Name | There is no graphical way to change this property. You can only change it in the msmdsrv.ini file. The XML node in the file is named CoordinatorCancelCount. |
| Default Value | 1000 |
| Unit of Measure | Internal Iterations |
| Data Type | Integer |
| Minimum Value | 1 – The coordinator will check for canceled queries upon each internal iteration. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | The default value of 1000 has been determined by Microsoft to be the correct balance between query performance and responsiveness to canceled queries. |
| Security Implications | None |

##### CoordinatorExecutionMode

**CoordinatorExecutionMode** is a complex SSAS server property that controls the number of parallel jobs that Analysis Server will start. When operations (such as querying and processing) occur in parallel, performance can be significantly enhanced. However, too many parallel jobs on a server can decrease responsiveness to other operations. The CoordinatorExecutionMode property’s value has special meaning as it controls the SSAS coordinator component, as follows:

• **Positive number.** Sets the absolute maximum number of parallel jobs per server.

• **Zero.** Enables the server to automatically determine the maximum number of parallel operations, based on workload and system resources.

• **Negative number.** Sets the absolute maximum number of parallel jobs that can occur per processor (as opposed to a positive number, which is per server).

|  |  |
| --- | --- |
| Property Name | General Page: CoordinatorExecutionMode |
| Default Value | -4 |
| Unit of Measure | Parallel Operations |
| Data Type | Integer |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Use caution when setting this property, as it can significantly decrease the performance of your SSAS server if you set these values too high (either positive or negative) for your particular server workload. |
| Security Implications | None |

##### DatabaseConnectionPoolMax

**DatabaseConnectionPoolMax** is a server property that controls the size of the connection pool. Opening connection to the relational database is expencive operation. It might take up to several tens of seconds to get a single relational database connection fully intitialized. Analysis Server will not close established connections immideately but will keep them in the pool till next operation comes.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Process \ DatabaseConnectionPoolMax |
| Default Value | 50 |
| Unit of Measure | Number of connection objects |
| Data Type | Integer |
| Minimum Value | 1 |
| Maximum Value | 2147483647 |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | Increase the property if many processing jobs run in parallel. Decreasing the value will cause SSAS to be more responsive in cases when databases referenced by connections are no longer available. |
| Security Implications | None |

##### DatabaseConnectionPoolConnectTimeout

**DatabaseConnectionPoolConnectTimeout** is a server property that is used to set the number of seconds that SSAS should wait to connect to relational data sources before timing out.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Process \ DatabaseConnectionPoolConnectTimeout |
| Default Value | 60 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 1 |
| Maximum Value | 2147483647 |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### DataDir

**DataDir** is a server property that specifies where Analysis Server data is stored per server instance.

|  |  |
| --- | --- |
| Property Name | General Page: DataDir |
| Default Value | Determined during setup, but is here by default:  *<drive>*:\Program Files\Microsoft SQL Server\*<Install Folder>*\OLAP\Data |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | This folder is specifically reserved for Analysis Services. It is not the same folder as the one used for the Database Engine.  Analysis Services does not check the validity of the path you enter for this server property. Ensure the path exists when changing this property.  You might want to plan ahead and place your data directory on a drive with sufficient space for growth.  Performance of the drive where the data directory is located will affect performance of Analysis Services. You might want to place the data directory on a high performance drive or create a disk array to improve I/O performance. |
| Security Implications | If you change this property, you may need to create the folder and also set the appropriate security permissions on it. |

##### DeepCompressValue

**DeepCompressValue** is a server property that controls the compression for double datatypes. When deep compression is used, double datatypes require less space for storage. However, this comes at the expense of less precision. When deep compression is selected, as it is by default, the 15th digit to the right of the decimal point is lost. If this level of precision is important to your data, set this value to 0. However, storage requirements will increase.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Process \ DeepCompressValue |
| Default Value | 1 |
| Unit of Measure | N/A |
| Data Type | Integer |
| Minimum Value | 0 – Indicates that no compression will be used on double datatypes, but full precision will be preserved. |
| Maximum Value | 1 – Indicates that compression will be used on double datatypes, but at the expense of losing the last digit of precision. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | If you change this value, you must reprocess all dimensions, cubes, and partitions to use the new setting. |
| Security Implications | None |

##### DefaultDrillthroughMaxRows

**DefaultDrillthroughMaxRows** is a server property that controls the maximum number of rows that can be returned from a drill-through query.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Query \ DefaultDrillthroughMaxRows |
| Default Value | 10000 |
| Unit of Measure | Rows |
| Data Type | Integer |
| Minimum Value | 0 – Maximum number of rows is not limited. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | If this value is set too high, SSAS can consume too many network, client, or server resources. Consider how many rows will need to be presented in your applications before changing this value. |
| Security Implications | None |

##### DSO - LocksDirectory

**DSO – LocksDirectory** is a server property that indicates which directory should be used by the legacy DSO object model when connecting to SSAS 2005. By default, the value of this property is empty. Users needing to enable legacy DSO applications to work against SSAS 2005 must set up DSO–related server properties. Along with the **RepositoryConnectionString** property, this property is required for the functioning of a DSO application on the local machine.

|  |  |
| --- | --- |
| Property Name | General Page: DSO \ LocksDirectory |
| Default Value | Empty |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | It is important to know that you should not modify or change objects created with legacy DSO applications by using SSAS management and development tools. The user needs to set up a SSAS 2000–style repository by either copying msmdrep.mdb from the SSAS 2000 installation folder or by setting up a repository migrated to SQL Server. |
| Security Implications | None |

##### DSO - RemoteLocksDirectory

**DSO – RemoteLocksDirectory** is a server property that indicates which directory should be used by the legacy DSO object model when connecting to SSAS 2005 and operating remotely from another machine. By default, the value of this property is empty. The property behavior is similar to **LocksDirectory**.

|  |  |
| --- | --- |
| Property Name | General Page: DSO \ RemoteLocksDirectory |
| Default Value | Empty |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | See special notes for the LocksDirectory server property. |
| Security Implications | None |

##### DSO - RemoteRepositoryConnectionString

**DSO – RemoteRepositoryConnectionString** is a server property that indicates how the legacy DSO object model should connect to the repository database when establishing a remote connection to SSAS 2005. By default, the value of this property is empty.

|  |  |
| --- | --- |
| Property Name | General Page: DSO \ RemoteRepositoryConnectionString |
| Default Value | Empty |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Populate this property only if remote legacy DSO applications require remote access to SSAS 2005. |
| Security Implications | DSO – RemoteRepositoryConnectionString is stored in encrypted format by Analysis Server. |

##### DSO - RepositoryConnectionString

**DSO – RepositoryConnectionString** is a server property that indicates which legacy DSO object model should connect to the repository database. By default, the value of this property is empty. You will need to set up a SSAS 2000–style repository by either copying msmdrep.mdb from the SSAS 2000 installation folder or by setting up a repository migrated to SQL Server and then specifying a connection string pointing to the repository.

|  |  |
| --- | --- |
| Property Name | General Page: DSO \ RepositoryConnectionString |
| Default Value | Empty |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | DSO – RepositoryConnectionString is stored in encrypted format by Analysis Server. |

##### EnableBinaryXML

**EnableBinaryXML** is a Boolean server property that dictates whether XML is to be streamed in a binary format. This property appears twice in the General page of the Analysis Server Properties page. It appears once under Network \ Requests and once under Network \ Responses. The default value for requests is **False** because requests are relatively short. The default value for responses is **True** because responses are generally large amounts of data. Requests and responses can also be compressed. See the **EnableCompression** server property. You can also override this server behavior by specifying **Protocol Format** connection string property.

|  |  |
| --- | --- |
| Property Name | General Page: Network \ Requests \ EnableBinaryXML  General Page: Network \ Responses \ EnableBinaryXML |
| Default Value | False for requests  True for responses |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – XML will not be streamed as binary. |
| Maximum Value | True – XML will be streamed as binary. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### EnableCompression

**EnableCompression** is a Boolean server property that dictates whether data by default is compressed when it goes across the wire. This property appears twice in the General page of the Analysis Server Properties page. It appears once under Network \ Requests and once under Network \ Responses. The default value for requests is **False** because requests are relatively short. The default value for responses is **True** because responses are generally large amounts of data. To control whether requests and responses are streamed in binary format, see the **EnableBinaryXML** server property. You can also override this server behavior by specifying **Transport Compression** connection string property.

|  |  |
| --- | --- |
| Property Name | General Page: Network \ Requests \ EnableCompression  General Page: Network \ Responses \ EnableCompression |
| Default Value | False for requests  True for responses |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – XML binary streams will not be compressed. |
| Maximum Value | True – XML binary streams will be compressed. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### ExternalCommandTimeout

**ExternalCommandTimeout** is a server property that is used to set the number of seconds that SSAS should wait to time out when issuing commands to external data sources, such as relational and other OLAP sources. For the timeout of connections to external sources, see **ExternalConnectionTimeout**.

|  |  |
| --- | --- |
| Property Name | General Page: ExternalCommandTimeout |
| Default Value | 3600 (one hour) |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 0 |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### ExternalConnectionTimeout

**ExternalConnectionTimeout** is a server property that is used to set the number of seconds, by default, that SSAS should wait to time out when connecting to external data sources, such as relational and other OLAP sources. For the timeout of actual commands to external sources, see **ExternalCommandTimeout**.

|  |  |
| --- | --- |
| Property Name | General Page: ExternalConnectionTimeout |
| Default Value | 60 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 0 |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | This property is ignored if the connection string specifies the connection timeout value. |
| Security Implications | None |

##### FlightRecorder - Enabled

**FlightRecorder - Enabled** is a Boolean server property that determines whether flight recorder trace is running. The concept of flight recorder trace is similar to “black box” recorders used to record in-flight activity of an airplane. Flight Recorder trace, when enabled, is constantly recording Analysis Server activity into a short term log, enabling SSAS administrators to troubleshoot problems. The Flight Recorder is logging activity into a FlightRecCurrent.trc file, which is located under the folder specified by the **LogDir** server property. After the amount of time specified by the **FlightRecorder – LogDurationSec** server property is passed, or whenever FlightRecCurrent.trc is reaching the size limit specified by the **FlightRecorder – FileSizeMB** server property, the FlightRecCurrent.trc file is renamed to FlightRecBack.trc and a new FlightRecCurrent.trc file is created. At the same moment, the old copy of FlightRecBack.trc is deleted. You can open FlightRecCurrent.trc or FlightRecBack.trc using the SQL Server Profiler application. You can also try and replay events captured by the flight recorder using the replay functionality of SQL Server Profiler.

One of the problems administrators will encounter when investigating a Flight Recorder trace is that some trace events will be associated with sessions and connections started before Analysis Server began logging into the current FlightRecCurrent.trc or FlightRecBack.trc.

In order for Flight Recorder to capture the Analysis Server state, currently opened connections, sessions, jobs, and active locks Analysis Server will connect to itself and issue several DISCOVER requests that are going to get logged into a Flight Recorder trace. The frequency of these DISCOVER requests is determined by the value set in the **FlightRecorder – SnapshotFrequencySec** server property. Which requests are being sent is determined by the file referenced in the **FlightRecorder – SnapshotDefinitionFile** server property.

The list of events logged by the Flight recorder trace is specified in a file referred to in the **FlightRecorder – TraceDefinitionFile** server property.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ FlightRecorder \ Enabled |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Flight recorder is not running. |
| Maximum Value | True – Flight recorder trace is logging recent Analysis Server activity. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | You might want to increase the value of FlightRecorder - LogDurationSec server property to log more information into flight recorder trace. Having more events logged into flight recorder trace might lead to performance degradation along with increasing the frequency of the snapshots. |
| Security Implications | None |

##### FlightRecorder - FileSizeMB

**FlightRecorder - FileSizeMB** is a server property that controls the size of the flight recorder file. For a general discussion about Flight Recorder trace, see the **FlightRecorder – Enabled** server property.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ FlightRecorder \ FileSizeMB |
| Default Value | 10 |
| Unit of Measure | Megabytes |
| Data Type | Integer |
| Minimum Value | 0 – Flight recorder trace will not log any events. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Make sure you have twice the free space specified by this property because Analysis Server keeps two files (FlightRecBack.trc and FlightRecCurrent.trc) in the folder specified by the LogDir server property |
| Security Implications | None |

##### FlightRecorder - LogDurationSec

**FlightRecorder - LogDurationSec** is a server property that specifies an upper limit on the duration of flight recorder trace. Along with **FlightRecorder – FileSizeMB**, this property allows you to control the size of the flight recorder trace. Flight recorder will start recording a new FlightRecCurrent.trc file whenever any of the conditions occur. For a general discussion of Flight Recorder trace, see the **FlightRecorder – Enabled** server property.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ FlightRecorder \ LogDurationSec |
| Default Value | 3600 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 0 – Flight recorder trace will not log any events. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Make sure you have twice the free space specified in the FlightRecorder – FileSizeMB server property because Analysis Server keeps two files (FlightRecBack.trc and FlightRecCurrent.trc) in the folder specified by the LogDir server property. |
| Security Implications | None |

##### FlightRecorder - SnapshotDefinitionFile

**FlightRecorder - SnapshotDefinitionFile** is a property that specifies the name of the file containing the definition of the DISCOVER requests that Analysis Server sends while recording Flight recorder trace. For a general discussion of Flight Recorder trace, see the **FlightRecorder – Enabled** server property.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ FlightRecorder \ SnapshotDefinitionFile |
| Default Value | Empty – By default, snapshot definitions are located in flightrecordersnapshotdef.xml, located at *<drive>*:\Program Files\Microsoft SQL Server\*<Install Folder>*\OLAP\bin. |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications |  |

##### FlightRecorder - SnapshotFrequencySec

**FlightRecorder - SnapshotFrequencySec** is a server property that specifies how often Analysis Server will connect to itself and issue a series of DISCOVER events (snapshots) to determine its own state. For a general discussion of Flight Recorder trace, see the **FlightRecorder – Enabled** server property.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ FlightRecorder \ SnapshotFrequencySec |
| Default Value | 120 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 1 – Snapshots are issued every second. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Frequent snapshots might cause some performance degradation. Having too few snapshots could make it harder to troubleshoot recent activity. |
| Security Implications | None |

##### FlightRecorder - TraceDefinitionFile

**FlightRecorder - TraceDefinitionFile** is a property that specifies what trace events are being recorded by Flight Recorder trace. For a general discussion of Flight Recorder trace, see the **FlightRecorder – Enabled** server property.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ FlightRecorder \ TraceDefinitionFile |
| Default Value | Empty – By default, snapshot definitions are located in flightrecordertracedef.xml, located at *<drive>*:\Program Files\Microsoft SQL Server\*<Install Folder>*\OLAP\bin. |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications |  |

##### ForceCommitTimeout

**ForceCommitTimeout** is a server property that is used to control what happens when a processing operation is waiting to finish its operation to enter the commit phase. When this value is greater than zero, SSAS will start canceling prior transactions, but only after the specified value in milliseconds. However, if read locks become available before the **ForceCommitTimeout** period is reached, canceling will not occur. See discussion of read locks under **CommitTimeout**.

|  |  |
| --- | --- |
| Property Name | General Page: ForceCommitTimeout |
| Default Value | 30000 (30 seconds) |
| Unit of Measure | Milliseconds |
| Data Type | Integer |
| Minimum Value | 0 – Will not force a commit timeout. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### IdleConnectionTimeout

**IdleConnectionTimeout** is a server property that controls the timeout of connections that have not been used for a specified amount of time. After timeout expires the connections will be discarded by Analysis Server. By default, idle connections will not timeout.

|  |  |
| --- | --- |
| Property Name | General Page: IdleConnectionTimeout |
| Default Value | 0 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 0 – Will not force a timeout of idle connections. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### IdleOrphanSessionTimeout

IdleOrphanSessionTimeout is a server property that controls the timeout of idle orphaned sessions. If the connection dies, the internal connection object is destroyed, but the session associated with the connection still lives and becomes an orphaned session. Analysis Services clients will by default create a session when establishing connection to Analysis Server.

The default timeout for orphaned sessions is 120 seconds. If a connection is not made to the orphaned session in this period of time, the session will be destroyed.

|  |  |
| --- | --- |
| Property Name | General Page: IdleOrphanSessionTimeout |
| Default Value | 120 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 0 – Will not force a timeout of orphaned sessions. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | An application needs to track the SessionID to be able to connect to the same session again. If your applications do not have a way to do this, ensure that you do not set this value to 0, or orphaned sessions will consume unnecessary resources indefinitely. |
| Security Implications | None |

##### InstanceVisible

**InstanceVisible** is a Boolean server property that flags whether a SSAS instance is made available for browsing when the instance reports itself to SQL Server Browser. If this property value is set to **False**, a connection must be made to the specific named instance using its specific port number. To learn about SSAS instances, see the “Instances” section at the beginning of this paper.

|  |  |
| --- | --- |
| Property Name | General Page: InstanceVisible |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – The instance is not available for connecting through SQL Browser. |
| Maximum Value | True – The instance is available for connecting through SQL Browser. |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | If this property is set to True, you must also have the SQL Server Browser service running to connect to the instance. |
| Security Implications | None |

##### IPV4Support

**IPV4Support** is a server property that controls when Analysis Server is going to use version 4 of Internet Protocol. Values accepted by this property are:

• **0.** IPv4 is disabled. Analysis Server doesn't listen on the TCP/IPv4 port, and clients won't be able to connect using IPv4.

• **1.** IPv4 is required. Analysis Server listens on TCP/IPv4. If the protocol is not available Analysis Server will not start.

• **2.** IPv4 is optional. Analysis Server tries to listen on IPv4, but will silently ignore errors and continue to start if IPv4 is not available.

|  |  |
| --- | --- |
| Property Name | General Page: Network \ Listener \ IPV4Support |
| Default Value | 1 |
| Unit of Measure | Enumeration |
| Data Type | Integer |
| Minimum Value | 0 |
| Maximum Value | 2 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### IPV6Support

**IPV6Support** is a server property that controls when Analysis Server is going to use version 6 of Internet Protocol. Version 6 is a new networking protocol that is not widely available. Values accepted by this property are:

• **0.** IPv6 is disabled. Analysis Server doesn't listen on the TCP/IPv6 port, and clients won't be able to connect using IPv6.

• **1.** IPv6 is required. Analysis Server listens on the TCP/IPv6 port. If the protocol is not available, Analysis Server will not start.

• **2.** IPv6 is optional. Analysis Server tries to listen on IPv6, but will silently ignore errors and continue to start if IPv6 is not available.

|  |  |
| --- | --- |
| Property Name | General Page: Network \ Listener \ IPV4Support |
| Default Value | 2 |
| Unit of Measure | Enumeration |
| Data Type | Integer |
| Minimum Value | 0 |
| Maximum Value | 2 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### Language

**Language** is a server property that dictates the default language used by all databases on a server. This setting is statically inherited to each new database, but can be overwritten by the database itself. See the “Languages” section at the beginning of this paper.

|  |  |
| --- | --- |
| Property Name | Language/Collation Page: Language |
| Default Value | Determined by SKU of installed product. |
| Unit of Measure | N/A |
| Data Type | String / Dropdown list |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### LazyProcessing - Enabled

**LazyProcessing - Enabled** is a Boolean server property that determines whether SSAS will be allowed to perform background processing of the indexes and aggregations, bringing partitions and dimensions to the optimized state. Partition indexes and aggregations might be dropped as a result of an update of the dimension, which causes significant slowdown in query performance.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ LazyProcessing \ Enabled |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – No lazy processing performed by SSAS. |
| Maximum Value | True – Lazy processing is allowed. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Lazy processing allows fast updates to the dimensions. After an update, SSAS processing of aggregations and indexes is initiated in the background. As a result of lazy processing, you might see Analysis Server consuming CPU and disk I/O at times when no user activity is performed. To avoid lazy processing, you can set the ProcessAffectedObjects property as part of a dimension update processing command, causing all aggregations and indexes to be processed as part of the same transaction. |
| Security Implications | None |

##### LazyProcessing - MaxObjectsInParallel

**LazyProcessing – MaxObjectsInParallel** is a server property that controls how many objects at a time are being processed during a lazy operation.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ LazyProcessing \ MaxObjectsInParallel |
| Default Value | 2 |
| Unit of Measure | Objects |
| Data Type | Integer |
| Minimum Value | 1 |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### LazyProcessing - SleepIntervalSecs

**LazyProcessing – SleepIntervalSecs** is a server property that controls how often a background lazy thread wakes up. A lazy processing thread wakes up according to the value set in this server property and iterates all objects on Analysis Server, looking for any object marked for lazy processing.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ LazyProcessing \ SleepIntervalSecs |
| Default Value | 5 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 1 - Lazy processing thread will wake up every second. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Tune this value to adjust how fast your lazy processing will start after a dimension update. |
| Security Implications | None |

##### LinkFromOtherInstanceEnabled

**LinkFromOtherInstanceEnabled** is a Boolean server property that dictates whether the current SSAS instance object can be linked from another SSAS instance. By default, another instance cannot link to the current instance objects. To allow the creation of linked objects in the current instance, see the **LinkToOtherInstanceEnabled** property.

|  |  |
| --- | --- |
| Property Name | General Page: Feature \ LinkFromOtherInstanceEnabled |
| Default Value | False |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Current instance objects cannot be linked from another instance. |
| Maximum Value | True – Current instance objects can be linked from another instance. |
| Requires Restart | No |
| Alternate GUI Tool | SAC/Features: Linked Objects \ Enable links from other instances |
| Special Notes | None |
| Security Implications | None |

##### LinkInsideInstanceEnabled

**LinkInsideInstanceEnabled** is a Boolean server property that dictates whether objects from different databases within the current SSAS instance can link to each other. By default, you can create linked objects from databases inside the current instance.

|  |  |
| --- | --- |
| Property Name | General Page: Feature \ LinkInsideInstanceEnabled |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Objects from different databases inside the current instance cannot link to each other. |
| Maximum Value | True – Objects from different databases inside the current instance can link to each other. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### LinkToOtherInstanceEnabled

**LinkToOtherInstanceEnabled** is a Boolean server property that dictates whether the current SSAS instance can create linked objects pointing to another SSAS instance. By default, the current instance cannot create linked objects pointing to another instance. To enable another instance to link to the current instance, see the **LinkFromOtherInstanceEnabled** property.

|  |  |
| --- | --- |
| Property Name | General Page: Feature \ LinkToOtherInstanceEnabled |
| Default Value | False |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Current instance cannot create linked objects pointing to another instance. |
| Maximum Value | True – Current instance can create linked objects pointing to another instance. |
| Requires Restart | No |
| Alternate GUI Tool | SAC/Features: Linked Objects \ Enable links to other instances |
| Special Notes | None |
| Security Implications | None |

##### ListenOnlyOnLocalConnections

**ListenOnlyOnLocalConnections** is a Boolean server property that dictates whether Analysis Server will listen to connections coming from remote machines or will only listen to locally originated connections. This property could be used to restrict remote access to Analysis Server during a security attack.

|  |  |
| --- | --- |
| Property Name | General Page: Network \ ListenOnlyOnLocalConnections |
| Default Value | False |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Remote connections are allowed. |
| Maximum Value | True – Only connections originating from local machine will be accepted by Analysis Server. |
| Requires Restart | Yes |
| Alternate GUI Tool | SAC/Services: Analysis Services \ Remote Connections |
| Special Notes | None |
| Security Implications | None |

##### LogDir

**LogDir** is a property that specifies where Analysis Server logs and traces are stored.

|  |  |
| --- | --- |
| Property Name | General Page: LogDir |
| Default Value | Determined during setup, but is here by default:  *<drive>*:\Program Files\Microsoft SQL Server\*<Install Folder>*\OLAP\Log |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | This folder is specifically reserved for Analysis Server. It is not the same folder used for the Database Engine.  Analysis Services does not check the validity of the path you enter for this server property. Ensure the path exists when changing this property. |
| Security Implications | If you change this property, you may need to create the folder and also set the appropriate security permissions on it. |

##### LowMemoryLimit

**LowMemoryLimit** is a server property that specifies the low threshold of physical memory available to the server. This value is expressed as a percentage of total memory if the value is less than 100, or an absolute value of bytes if the value is greater than 100. For more information on memory, see the “Memory” section of this paper.

|  |  |
| --- | --- |
| Property Name | General Page: Memory \ LowMemoryLimit |
| Default Value | 75 |
| Unit of Measure | Percentage if between 0 and 100.  Bytes if greater than 100. |
| Data Type | Double |
| Minimum Value | 0 – SSAS cleaner will be allowed to clean any cached query results. |
| Maximum Value | For Percentage: 100 – SSAS will use up to 100% of total physical memory.  For Bytes: 1.79E + 308 is the maximum value that can be specified. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### MaxConcurrentPredictionQueries

**MaxConcurrentPredictionQueries** is a server property that specifies the maximum number of prediction queries that can be run simultaneously. The default value is 0, which does not limit the number of prediction queries.

|  |  |
| --- | --- |
| Property Name | General Page: DataMining \ MaxConcurrentPredictionQueries |
| Default Value | 0 |
| Unit of Measure | Number of queries |
| Data Type | Integer |
| Minimum Value | 0 – An unlimited number of prediction queries can be run simultaneously. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### MaxIdleSessionTimeout

**MaxIdleSessionTimeout** is a server property that controls the timeout of sessions that are idle. If this property is not set to a value of 0, sessions that exceed this value are automatically destroyed. Under no circumstances will an idle session live longer than the value specified in this property, but it can live shorter. See **MinIdleSessionTimeout** for more information.

|  |  |
| --- | --- |
| Property Name | General Page: MaxIdleSessionTimeout |
| Default Value | 0 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 0 – Will not force a timeout of idle sessions. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### MaxThreads

**MaxThreads** is a server property that controls how many threads will be spawned, at maximum for either query or process operations.

|  |  |
| --- | --- |
| Property Name | General Page: ThreadPool \ Process \ MaxThreads  General Page: ThreadPool \ Query \ MaxThreads |
| Default Value | 10 for Query  64 for Process |
| Unit of Measure | Threads |
| Data Type | Integer |
| Minimum Value | 1 – A maximum of one thread is spawned. |
| Maximum Value | 2147483647, although this number is not practical for threading. The practical number is determined by the operating system. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | This is server-wide setting that affects all queries and processing operations.  Regardless of the number of processors in a server, the default values apply. If your server has more than one processor, you should increase the value. |
| Security Implications | None |

##### MemoryLimitErrorEnabled

**MemoryLimitErrorEnabled** is a Boolean server property that controls whether an error will be generated if the amount of estimated memory required cannot be granted. The default value for this property is **True**, which indicates that limit errors for estimated memory will be reported.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ ProcessPlan \ MemoryLimitErrorEnabled |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Limit errors for estimated memory will not be reported. |
| Maximum Value | True – Limit errors for estimated memory will be reported. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### MinIdleSessionTimeout

**MinIdleSessionTimeout** is a server property that controls the timeout of sessions that are idle, along with **MaxIdleSessionTimeout**. If this value is not set to 0, sessions which are idle for a period of time that is greater than the value specified in **MinIdleSessionTimeout** (but less than the **MaxIdleSessionTimeout**) can be automatically destroyed if needed by server resources.

|  |  |
| --- | --- |
| Property Name | General Page: MinIdleSessionTimeout |
| Default Value | 2700 |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 0 – Will not force a session to live for a minimum period of time. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### MinThreads

**MinThreads** is a server property that controls the minimum number of threads that will be spawned for either query or process operations.

|  |  |
| --- | --- |
| Property Name | General Page: ThreadPool \ Process \ MinThreads  General Page: ThreadPool \ Query \ MinThreads |
| Default Value | 1 for Query and Process |
| Unit of Measure | Threads |
| Data Type | Integer |
| Minimum Value | 1 – A minimum of one thread is spawned. |
| Maximum Value | 2147483647, although this number is not practical for threading. The practical number is determined by the operating system. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | This is a server-wide setting that affects all queries and processing operations. Increase the value to make more operations run in parallel. |
| Security Implications | None |

##### Port

**Port** is a server property that controls which port a SSAS instance listens on. The default instance listens on port 2383, while named instances are dynamically allocated. This behavior can be changed by explicitly assigning a port to either the default or named instance of SSAS. If the value for **Port** is 0, the default behavior described above is achieved. Otherwise, the value is treated as an absolute (static) port number to use for the specified instance. For a discussion about ports, see the “Instances” section of this paper.

|  |  |
| --- | --- |
| Property Name | General Page: Port |
| Default Value | 0 |
| Unit of Measure | Port number |
| Data Type | Integer |
| Minimum Value | 0 – Will use the default port configuration of 2383 for the default instance and will dynamically allocate a port for a named instance. |
| Maximum Value | 2147483647 |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | It is a good practice to allow SSAS to dynamically assign ports to named instances because it will not be the same port number when the service is restarted. However, if you are not using the SQL Server Browser service, you will likely have to assign a static port so that you know how to connect to the instance.  Please note that assigning ports should be part of the planning for your security infrastructure. You should be aware of all software and hardware firewals and filters controlling protocols in your network. |

##### QueryLogConnectionString

**QueryLogConnectionString** is a server property that specifies how Analysis Server connects to a relational database to collect query statistics. If a value in the **QueryLogConnectionString** property is specified, SSAS will try to log query statistics into a table specified by the **QueryLogTableName** server property. Statistics about query usage are later used during Usage Based Optimization to help you to design aggregations that better fit a usage pattern for one or more partitions. For more information about the SSAS query log, see the article “[Configuring the Analysis Services Query Log](http://go.microsoft.com/fwlink/?linkid=66164)”.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ QueryLog \ QueryLogConnectionString |
| Default Value | Empty |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | Query Log Connection string is stored in an encrypted format by Analysis Server. |
| Security Implications | Make sure the Analysis Server account has sufficient privileges to create a table in the database referenced by the QueryLogConnectionString property. Set the CreateQueryLogTable server property value to True to enable the table to be created. After the table is created by Analysis Server, only privileges to insert into and delete from the table are required. It is recommended to create a separate database to collect query statistics to minimize the security attack surface. |

##### QueryLogSampling

**QueryLogSampling** is a server property that specifies how often queries are being logged into the query log table by Analysis Server. If QueryLogSampling is set to 1, every query is logged. By default, only every tenth query is logged. For more information about the SSAS query log, see the article “[Configuring the Analysis Services Query Log](http://go.microsoft.com/fwlink/?linkid=66164)”.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ QueryLog \ QueryLogSampling |
| Default Value | 10 |
| Unit of Measure | Frequency of queries logged |
| Data Type | Integer |
| Minimum Value | 1 – Every query is going to be logged into a query log. |
| Maximum Value | 2147483647 – Only when the counter reaches maximum value for an integer value, the information for a single query is logged into a query log table. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | Query Log Connection string is stored in encrypted format by Analysis Server. |
| Security Implications | None |

##### QueryLogTableName

**QueryLogTableName** is a server property that specifies the relational database table name for Analysis Server to log statistical information about queries. Make sure you specify values for the QueryLogConnectionString and CreateQueryLogTable server properties to True in order to enable SSAS to start logging. For more information about the SSAS query log, see the article “[Configuring the Analysis Services Query Log](http://go.microsoft.com/fwlink/?linkid=66164)”.

|  |  |
| --- | --- |
| Property Name | General Page: Log \ QueryLog \ QueryLogConnectionString |
| Default Value | Empty |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### RequireClientAuthentication

**RequireClientAuthentication** is a Boolean server property that controls whether anonymous connections are allowed to SSAS. The default value for this property is **True**, which indicates that client authentication is required (or anonymous connections are disallowed).

|  |  |
| --- | --- |
| Property Name | General Page: Security \ RequireClientAuthentication |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – Client authentication is not required, which allows anonymous connections to SSAS. |
| Maximum Value | True – Client authentication is required, which disallows anonymous connections to SSAS. |
| Requires Restart | No |
| Alternate GUI Tool | SAC/Features: Anonymous Connections \ Enable anonymous connections |
| Special Notes | The GUI tool sets the property opposite of the RequireClientAuthentication server property. In other words, setting Enable anonymous connections to True, sets the RequireClientAuthentication server property to False and vice-versa. |
| Security Implications | Allowing anonymous connections could pose a security risk. |

##### ROLAPDimensionProcessingEffort

**ROLAPDimensionProcessingEffort** is a server property that controls the maximum number of rows SSAS will obtain from a relational database in an attempt to resolve a query to a ROLAP dimension. If Analysis Server determines that the number of records that must be processed from the underlying database is greater than the value specified by this property, an error is returned to the client.

|  |  |
| --- | --- |
| Property Name | General Page: OLAP \ Process \ ROLAPDimensionProcessingEffort |
| Default Value | 300000 |
| Unit of Measure | Data records |
| Data Type | Integer |
| Minimum Value | 0 – No query to a ROLAP dimension will succeed. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### ServerTimeout

**ServerTimeout** is a server property that controls the timeout of queries that are issued against SSAS. The default value is 3600, which indicates 60 minutes.

|  |  |
| --- | --- |
| Property Name | General Page: ServerTimeout |
| Default Value | 3600 (one hour) |
| Unit of Measure | Seconds |
| Data Type | Integer |
| Minimum Value | 0 – Indicates that Analysis Services will wait indefinitely for queries to finish. |
| Maximum Value | 2147483647 |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

##### ServiceAccountIsServerAdmin

**ServiceAccountIsServerAdmin** is a Boolean server property that controls whether the SQL Server Analysis Services service itself is considered to have administrative privileges to Analysis Server. The default value for this property is **True**.

|  |  |
| --- | --- |
| Property Name | General Page: Security \ ServiceAccountIsServerAdmin |
| Default Value | True |
| Unit of Measure | N/A |
| Data Type | Boolean |
| Minimum Value | False – The SQL Server Analysis Services service does not run as a local administrator for SSAS. |
| Maximum Value | True – The SQL Server Analysis Services service runs as a local administrator for SSAS. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | Setting this server property to False might not give the service account enough privileges to function properly. |
| Security Implications | Any service running as an administrator could pose a security risk. |

##### TempDir

**TempDir** is a property that specifies where Analysis Server places temporary files that it uses during processing operations.

|  |  |
| --- | --- |
| Property Name | General Page: TempDir |
| Default Value | Determined during setup, but if not specified, this directory is the same as the value specified in the DataDir folder. |
| Unit of Measure | N/A |
| Data Type | String |
| Minimum Value | N/A |
| Maximum Value | N/A |
| Requires Restart | Yes |
| Alternate GUI Tool | None |
| Special Notes | Analysis Services does not check the validity of the path you enter for this server property. Ensure the path exists when changing this property. |
| Security Implications | If you change this property, you may need to create the folder and also set the appropriate security permissions on it. |

##### TotalMemoryLimit

**TotalMemoryLimit** is a server property that controls the total amount of physical memory that can be used by all SSAS operations. This value is expressed as a percentage of total memory if the value is less than 100, or an absolute value of bytes if the value is greater than 100. The default value is 80, which indicates that a maximum of 80 percent of the total physical memory can be used by SSAS. For more information on memory, see the “Memory” section at the beginning of this paper. The total amount of memory available to Analysis Server depends on the hardware platform Analysis Server runs on. A 32-bit operating system limits Analysis Server to maximum of 3 gigabytes (GB) with /Gb switch enabled in the boot.ini file. On a 64-bit platform, Analysis Server can address all memory available.

|  |  |
| --- | --- |
| Property Name | General Page: Memory \ TotalMemoryLimit |
| Default Value | 80 |
| Unit of Measure | Percentage if between 0 and 100.  Bytes if greater than 100. |
| Data Type | Double |
| Minimum Value | 0 – SSAS will not use any memory for caching. |
| Maximum Value | For Percentage: 100 – SSAS will use up to 100% of total physical memory.  For Bytes: 1.79E + 308 is the maximum value that can be specified. |
| Requires Restart | No |
| Alternate GUI Tool | None |
| Special Notes | None |
| Security Implications | None |

#### Conclusion

SQL Server Analysis Services (SSAS) server properties can be modified to enable you to configure SSAS to suit your own environment. These server properties control everything from memory to file locations, timeouts, features, and more. Setting these properties can be done using the SSAS property pages, but some can actually be set in the Surface Area Configuration Manager as well. This white paper presented the most commonly altered server properties and discussed (where appropriate) the ramifications of changing those server properties and the scenarios under which you many want to change them.

For more information:

<http://www.microsoft.com/technet/prodtechnol/sql/default.mspx>