

IBM Planning Analytics
Version 2 Release 0

TM1 Operations



Note

Before you use this information and the product it supports, read the information in [“Notices” on page 149](#).

Product Information

This document applies to IBM Planning Analytics Version 2.0 and might also apply to subsequent releases.

Licensed Materials - Property of IBM

Last updated: 2020-10-08

© **Copyright International Business Machines Corporation 2007, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Introduction.....	ix
Chapter 1. TM1 system and feature configuration.....	1
Multi-threaded queries.....	2
Capability assignments.....	3
Setting capabilities.....	5
The interaction of assignments in capability assignments.....	5
TM1 Application Web capabilities.....	5
Persistent feeders.....	6
Saving data with persistent feeders.....	6
Modifying rules for cubes with feeders.....	6
Handling corrupted or invalidated feeder files.....	6
Sandboxes and personal workspaces.....	7
Memory usage for sandboxes and personal workspaces.....	7
Managing files and folders for sandboxes.....	7
TM1 Server configuration parameters for sandboxes.....	7
Understanding sandbox differences among different TM1 clients.....	8
Understanding Recalc and Commit in a personal workspace.....	8
Chapter 2. TM1 Admin Server operations.....	9
Running the TM1 Admin Server on Windows.....	10
Running the TM1 Admin Server as an application on Windows.....	10
Running the TM1 Admin Server on UNIX.....	11
Specifying the location of the Admin Host.....	12
Chapter 3. Remote TM1 Server operations.....	13
Setting up a remote TM1 Server to run as an application.....	13
Starting a TM1 Server set up as a Windows application.....	14
Shutting down a TM1 Server set up as a Windows application.....	14
Setting up a TM1 Server to run as a Windows service.....	14
Starting a TM1 Server installed as a Windows service.....	15
Setting up a service to start automatically.....	15
Removing a TM1 Server from running as a Windows service.....	15
Shutting down a TM1 Server running as a Windows service.....	16
Starting a UNIX TM1 Server.....	16
Modifying the startup_tm1s.sh script.....	16
Running the UNIX TM1 Server in background mode.....	17
Shutting down a UNIX TM1 Server.....	17
Shutting down a UNIX TM1 Server running in background mode.....	17
Connecting to a remote server.....	18
Refreshing the list of remote servers.....	18
Resetting the Admin Host.....	19
Resetting local server options.....	19
Disconnecting from a remote server.....	19
Logging transactions.....	19
Monitoring server performance.....	19
Managing client connections.....	20
Broadcasting messages to clients.....	20
Disconnecting clients from the server.....	20
Shutting down a server remotely.....	21

Remote server memory management.....	21
About Stargate views.....	21
Using TM1 in bulk load mode.....	22
Considerations for using bulk load mode.....	22
Starting bulk load mode.....	22
Ending bulk load mode.....	23
TM1 C API.....	23
TurboIntegrator process commands.....	23
Chapter 4. Planning Analytics security overview.....	25
Overview to authentication.....	25
TM1 object security.....	25
Chapter 5. Managing users and groups.....	27
Users and groups security overview.....	27
Implementing a TM1 security scheme.....	27
User and group security examples.....	28
Administrative groups and authority.....	28
ADMIN group.....	29
SecurityAdmin group.....	29
DataAdmin group.....	30
OperationsAdmin group.....	30
Adding and deleting users and groups.....	32
Adding a user.....	32
Adding a group.....	33
Deleting a user.....	33
Deleting a group.....	33
Assigning users to groups.....	33
Membership in multiple groups.....	34
Securing TM1 data.....	34
Restricting access to the data directory.....	34
Assigning passwords.....	34
Securing passwords.....	34
Setting and clearing passwords.....	34
Setting a password.....	34
Clearing a password.....	35
Changing a password.....	35
Setting an expiration for a user.....	35
Chapter 6. System and performance monitoring.....	37
Planning Analytics and TM1 logs.....	37
The logging properties file.....	38
The TM1 Admin Server log.....	42
The transactions log.....	44
The message log.....	48
The audit log.....	50
The ops logger.....	55
The TM1 Top log.....	56
The TM1.Mdx.Interface logger.....	58
Searching large transaction logs.....	59
Control cubes.....	59
Enabling performance monitoring.....	59
Disabling performance monitoring.....	60
Viewing performance statistics for clients, cubes, and servers.....	60
TM1 performance counters.....	60
Available TM1 performance counters.....	61
Viewing TM1 performance counters with the TM1 PerfMon utility.....	61

Viewing performance counters with the Microsoft Windows Performance Monitor.....	62
--	----

Chapter 7. IBM Planning Analytics tools and utilities..... 65

Using the IBM Cognos TM1 Operations Console.....	65
IBM Cognos TM1 Operations Console workflow.....	66
Configuring the Cognos TM1 Operations Console.....	66
Starting and logging into the Cognos TM1 Operations Console.....	67
Setting up the Cognos TM1 Operations Console.....	67
Monitoring in the Cognos TM1 Operations Console.....	69
Log files in the IBM Cognos TM1 Operations Console.....	74
Using Watchdog to monitor server activity.....	79
IBM Cognos Performance Management Hub.....	80
Opening the Performance Management Hub web page.....	80
Configuring logging.....	81
Using the TM1 Top utility.....	81
Displaying user names in TM1 Top.....	82
Configuring the Tm1top.ini File.....	82
Running TM1 Top.....	83
Viewing and understanding the TM1 Top display.....	84
TM1 Top commands.....	88
Canceling a thread's processing.....	89
Other debugging tools.....	90
Installing and running ADPlus.....	90
Running Dr. Watson.....	93
tm1xfer.....	94
Collecting files for analysis following a TM1 Server crash.....	100
Client logging.....	100
odbc_test tool.....	101

Chapter 8. Understanding cube dependency.....103

Changes in dependency mapping.....	103
Clearing cube dependencies.....	104
Creating cube dependencies.....	104
Using AddCubeDependency to establish dependencies	105
Debugging and identifying cube dependencies.....	105

Chapter 9. Replicating cubes.....107

Relationships created by replication.....	108
Required access privileges.....	110
Set up replication.....	111
Configure the tm1s.cfg file to support replication.....	111
The replication process.....	112
Replicate a cube.....	112
Specifying information about the replicated cube.....	112
What happens when TM1 replicates a cube.....	114
The synchronization process.....	114
Synchronize on demand.....	115
Schedule synchronization.....	115
Synchronize over unstable or wide area network connections.....	116
Maintaining replication connections.....	116
Create a replication connection.....	116
Modify a replication connection.....	116
Delete a replication connection.....	117

Appendix A. Accessibility features..... 119

Keyboard shortcuts.....	119
IBM and accessibility.....	120

Appendix B. Control Cubes.....121

Security Control Cubes..... 121
 }CellSecurity_CubeName..... 121
 }ChoreSecurity.....122
 }ClientSecurity.....122
 }CubeSecurity.....122
 }DimensionSecurity.....123
 }ElementSecurity_DimensionName..... 123
 }ProcessSecurity..... 124
Client and Group Administration Control Cubes.....124
 }ClientsGroups..... 124
 }ClientProperties..... 125
Object Attribute and Property Control Cubes.....126
 }ConnectionProperties.....126
 }CubeProperties.....126
 }DimensionProperties.....127
 }DimensionAttributes.....127
 }ElementAttributes_DimensionName.....127
 }HierarchyProperties.....128
Performance Monitoring Control Cubes.....128
 }StatsByClient.....129
 }StatsByCube.....129
 }StatsByCubeByClient.....130
 }StatsByRule.....131
 }StatsForServer.....132
Other Control Cubes.....132
 }Hold_UserName_CubeName.....132

Appendix C. Control Dimensions..... 135

}Chores.....135
}ClientProperties.....135
}Clients.....135
}ConnectionProperties.....135
}Connections.....136
}CubeFunctions.....136
}CubeProperties.....137
}Cubes.....138
}DimensionAttributes.....139
}DimensionProperties.....139
}Dimensions.....140
}ElementAttributes_DimensionName.....140
}Groups.....140
}Hierarchies.....140
}HierarchyProperties.....140
}Hold.....141
}LineNumber.....141
}PerfClients.....141
}PerfCubes.....142
}Processes.....142
}RuleStats.....142
}StatsStatsByClient.....142
}StatsStatsByCube.....143
}StatsStatsByCubeByClient.....144
}StatsStatsForServer.....144
}Subsets_DimensionName.....144
}TimeIntervals.....145

}Views_CubeName.....	145
Appendix D. Excel events handled by TM1.....	147
Notices.....	149
Index.....	153

Introduction

This document describes IBM® Planning Analytics architecture, server operations, authentication, and component security.

Planning Analytics integrates business planning, performance measurement and operational data to enable companies to optimize business effectiveness and customer interaction regardless of geography or structure. Planning Analytics provides immediate visibility into data, accountability within a collaborative process and a consistent view of information, allowing managers to quickly stabilize operational fluctuations and take advantage of new opportunities.

Finding information

To find documentation on the web, including all translated documentation, access [IBM Knowledge Center](http://www.ibm.com/support/knowledgecenter) (<http://www.ibm.com/support/knowledgecenter>).

Samples disclaimer

The Sample Outdoors Company, Great Outdoors Company, GO Sales, any variation of the Sample Outdoors or Great Outdoors names, and Planning Sample depict fictitious business operations with sample data used to develop sample applications for IBM and IBM customers. These fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values is coincidental. Other sample files may contain fictional data manually or machine generated, factual data compiled from academic or public sources, or data used with permission of the copyright holder, for use as sample data to develop sample applications. Product names referenced may be the trademarks of their respective owners. Unauthorized duplication is prohibited.

Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products. Planning Analytics has components that support accessibility features.

For more information, see [Accessibility](#).

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of features or functionality remain at the sole discretion of IBM.

Security considerations

For security considerations for IBM Planning Analytics, see *Planning Analytics Installation and Configuration*. Information on managing user and group authentication can be found in the *Managing Users and Groups* chapter of the *TM1 Operations* documentation.

Chapter 1. TM1 system and feature configuration

You can use the following components and features to configure the IBM TM1® Server and clients for maximum performance.

Configuration	Description
Tm1s.cfg file	<p>A configuration file that specifies environment information for the TM1 Server.</p> <p>You can edit the Tm1s.cfg file to reflect the environment of the associated server.</p> <p>For more information, see <i>Planning Analytics Local Installation and Configuration</i>.</p>
Tm1p.ini file	<p>A configuration file that specifies the environment information for the TM1 clients (TM1 Perspectives, TM1 Architect, and Client).</p> <p>For more information, see <i>Planning Analytics Local Installation and Configuration</i>.</p>
Multi-threaded Queries	<p>Multi-threaded queries can calculate multiple streams of results in parallel to improve query performance. Multi-threaded queries allow IBM Cognos TM1 to automatically load balance the application of cores by running each query on a separate core. This multi-processing can improve efficiency and processing time for large queries and rules.</p> <p>For more information, see “Multi-threaded queries” on page 2.</p>
Capability Assignments	<p>A set of capabilities that administrators can enable or disable by user group.</p> <p>You can use capability assignments to manage options such as Personal Workspace Writeback Mode, Sandboxes, Data Reservation, and access to Server Explorer.</p> <p>For more information, see “Capability assignments” on page 3.</p>
Persistent Feeders	<p>A parameter that can improve reload time of cubes with feeders, especially feeders with many complex calculations, by saving feeders and then reloading them at server startup without recalculating those feeders.</p> <p>For more information, see “Persistent feeders” on page 6.</p>
Sandboxes	<p>A feature that allows users to create and work in their own data areas separate from base data.</p> <p>Administrator tasks for this feature include understanding the feature and its memory usage, enabling the necessary configuration parameters and managing related files and folders.</p> <p>For more information, see “Sandboxes and personal workspaces” on page 7.</p>

Configuration	Description
Data Reservation	<p>A server-related feature that you can use to configure exclusive write-access to regions of a cube for individual users. When it is reserved, the data in that region can be modified only by that specific user until the reservation is released.</p> <p>Administrator tasks for DR include configuring related properties in the }CubeProperties control cube and the Capability Assignments window.</p> <p>To apply and manage Data Reservations, you must use the related TurboIntegrator and API functions to programmatically obtain, release, and manage reservations.</p> <p>For more information, see the "Using Data Reservations" and "Enabling Data Reservation" sections in <i>TM1 for Developers</i>.</p>

Multi-threaded queries

You can improve the processing performance of queries by allowing queries to be split into multiple processing threads.

Multi-threaded queries allow IBM TM1 to automatically load balance the application of cores by executing each query on a separate core. This multiple processing can improve efficiency and processing time for large queries and rules.

Example

- Query 1 runs and receives the parent thread and the 7 worker threads (8 total).
- Query 2 runs. Upon completing the sub task in the worker thread, query 2 returns to be queued, and 4 of the threads are now assigned to Query 2, resulting in a 4 to 4 split of threads.
- Query 3 runs. The threads once again complete their sub task on the threads that are inflight, and then reassign to the new query. That causes a split of Query 1 into 3 threads, Query 2. Query 3 into 3 threads and Query 3 into 2 threads for a total of 8 threads. This process continues until eventually all threads are evenly divided.
- If Query 1 completes while Query 2 and Query 3 are still inflight, the threads that are assigned to Query 1 are assigned to Query 2 and Query 3, so they each maintain 4 threads.

Details

By default, multi-threaded queries are enabled. The number of threads is set to the maximum number of cores available on a server. The result is a dynamic system setting that consumes all cores. You can fine-tune the configuration by using the following parameters in the `tm1s.cfg` file:

- `MTQ`: Sets the maximum number of threads per TM1 query.
- `MTQ.OperationProgressCheckSkipLoopSize`: Specifies the number of cells to be processed before checking whether multi-threaded splits are needed.
- `MTQ.SingleCellConsolidation`: Disallow multi-threaded query processing for single cell consolidations.

For information about these parameters, see "Parameters in the `tm1s.cfg` File" in *Planning Analytics Local Installation and Configuration*.

Only servers running at less than 100% capacity can benefit from multi-threaded query improvements. For example, if you had 8 cores processing 8 concurrent queries, none could leverage multi-threaded queries. The addition of a 9th concurrently processing query would result in the query processing threads being split across multiple cores.

Multi-threaded queries can improve performance on numeric cubes, where consolidation is optimized. As TM1 does not consolidate string values, multi-threaded queries have no impact on the performance of string cubes.

Note: TM1 server multi-threaded queries are no longer limited to 64 CPU cores. Microsoft Windows Platforms with more than 64 processors can take full advantage of this feature and its performance benefits.

Logging

To enable logging for multi-threaded queries, enter the following lines in the `tm1s-log.properties` file located in the same location as your `tm1s.cfg` file:

- To capture Stargate creation times add: `log4j.logger.TM1.Cube.Stargate=DEBUG`
- To capture work unit splitting: `log4j.logger.TM1.Parallel=DEBUG`
- To capture the event of operation threads picking work units:
`log4j.logger.TM1.OperationThread=DEBUG`

Capability assignments

Administrators can set certain capabilities by user group using the Capability Assignment menu found in the **Server Explorer > Server Menu**.

With the Server Explorer open, right-click a server then click **Capability Assignments**.

By default all assignments are blank. The blank settings have different implications for each capability. Changes made to most assignments take effect after you log out of TM1 then log back in. The **Allow Export as Text** capability is dynamic; when you set it, you do not need to restart the TM1 Server.

The following capabilities can be set per user group:

Access to Server Explorer

Use this capability to launch the Server Explorer by default for this user group. You can set this capability to Deny to prevent the Server Explorer from being used by this user group. By default, this capability is set to blank, which is the equivalent of **Grant**. To block access to the Server Explorer, click the intersection of the user group and the capability and select **Deny**.

Personal Workspace Writeback Mode

This capability defines how data changes are handled in this user group.

When this capability is granted, users have the ability to hold data changes in a private workspace before manually deciding when to commit the changes to the base data. Users can further be granted the ability to name and manage multiple private scenarios called sandboxes (see the Sandbox Capability).

When Personal Workspace Writeback Mode is granted:

- Users operate in a non-direct writeback situation so they can privately adjust data values before making them available to the rest of the community. The special sandbox used when Personal Workspace Writeback Mode is granted makes it easy to try out different data changes without the complexity of named sandboxes.
- Data that is changed displays in a different color to remind the user that this change has not yet been merged to the base data. Once the data change in a Personal Workspace is committed, the cell coloring reverts to black to identify it as part of the base data.
- Users must manually commit their data changes to make them available to other users. If users are not used to working in a sandbox, for example, if they were used to the classic direct writeback mode of earlier versions of TM1, they may find it challenging to remember to commit their changes manually by pressing the Commit button.
- Often working in Personal Workspace mode improves performance over working in direct writeback since changes to the base data occur less frequently than when every data change must be merged to the base.

When this capability is set to **Deny**, the users do not have a Personal Workspace writeback so they work directly in the base. This is the default behavior for this capability. The advantage to this capability setting is that data changes happen immediately, but many users want the flexibility to control when to commit their changes and make them available to other users.

By default, user groups do not have the Personal Workspace Writeback mode capability granted. By default this capability is blank which acts like **Deny**.

To enable a user group to use a Personal Workspace, click the intersection of the user group and this capability and select **Grant**.

Note also that user groups may Deny the Personal Workspace Writeback Mode capability but still use the Sandbox capability. In that case, users work directly in the base but have the option to save changes to a named sandbox. When they create a named sandbox, then the Commit and reset Data buttons become available. If they move back to the Base, they return to direct writeback. See [“Setting capabilities” on page 5](#) for the possible option combinations of Personal Workspace Writeback Mode and Sandbox capabilities.

If the system-wide ability to have sandboxes is turned off using the DisableSandboxing=T in the server configuration file, the Personal Workspace Writeback Mode and Sandbox capabilities are ignored. By default DisableSandboxing is not present or is set to F.

Sandbox

This capability enables the user group to create named sandboxes that can be used to build what-if scenarios. With this capability granted, users can create and name more than one set of data changes which are overlaid on top of the base data. Users can keep their sandboxes private and decide when to commit the data changes to make them public. Sandboxing is a powerful feature, but some users will find the complexity of managing multiple scenarios against a common base challenging.

By default all users have this capability. The blank setting acts like **Grant**.

To prevent this user group from using named sandboxes, set this capability to **Deny**.

If the ability to have sandboxes is turned off using the DisableSandboxing configuration parameter in the server's configuration file, the Personal Workspace Writeback Mode and Sandbox capabilities are ignored. By default this parameter is not present or is set to F. DisableSandboxing is described in the *Planning Analytics Local Installation and Configuration* documentation.

Data Reservation capabilities

Data Reservation (DR) is a server-related feature that allows you to configure exclusive write-access to regions of a cube for individual users. Once reserved, the data in that region can only be modified by that specific user until the reservation is released.

DR uses the **ManageDataReservation** and **DataReservationOverride** capabilities to allow members of a user group to acquire, release, and override DRs for themselves and other users.

For details, see the "Using Data Reservations" and "Enabling user groups to manage Data Reservations" sections in the *TM1 for Developers* documentation.

Consolidation TypeIn Spreading

Consolidation TypeIn Spreading grants or denies the ability to perform data spreading by typing directly in a consolidated cell in the TM1 web-based clients, including TM1 Web and TM1 Application Web. If this capability is set to Deny, users are prevented from typing in a consolidated cell to initiate a data spread. However, when Consolidation TypeIn Spreading is set to **Deny**, users can still use the right-click menu to perform data spreading on a consolidated cell.

Note: This capability does not apply to the Cube Viewer in Architect or Perspectives, as typing directly in a consolidated cell is not allowed in these clients.

This capability cannot be set to **Deny** for the ADMIN user group.

A blank setting is equivalent to **Grant**.

Allow Spreading

Allow Spreading grants or denies access to all data spreading capabilities in any of the TM1 clients.

A blank setting is equivalent to **Grant**.

Allow Export as Text

Allow Export as Text grants or denies the ability to export cube or view data as text. Exporting a large cube or view can consume significant resources, so you may want to deny this capability to some user groups.

When this capability is set to **Deny**, members of the user group cannot export data as text and the following limitations are enforced:

- The **Export as Text Data** option is disabled when a user right-clicks on a cube or view in Server Explorer .
- An attempt to execute a TurboIntegrator process containing either the ASCIIOutput or TextOutput function results in the process exiting with a permission error. The process message log indicates "Execution was aborted. No security access for ASCIIOutput" or "Execution was aborted. No security access for TextOutput".

When this capability is set to **Grant**, members of the user group can export data as text.

A blank setting is equivalent to **Grant**.

Setting capabilities

You determine the writeback mode using a combination of Grant or Deny settings to the Personal Workspace Writeback Mode and the Sandbox Capabilities.

To let the usergroup	Personal Workspace Writeback Mode	Sandbox
Work directly in the base with the ability to create multiple sandboxes. This is the default case.	Deny	Grant
Work in a single, private work area without named sandboxes.	Grant	Deny
Work with both a "default" private work area and named sandboxes.	Grant	Grant
Work in Direct writeback with no sandbox. This is the classic TM1 mode. This setting is similar to the addition of DisableSandboxing=T to the configuration file which prevents the use of sandboxes and puts all users into direct writeback.	Deny	Deny

The interaction of assignments in capability assignments

By default all Capability Assignments are set to blank, which means either Deny or Grant depending on the default setting of each Capability.

For example, Blank in the Personal Workspace Writeback Mode Capability means Deny. Blank in the Sandbox Capability means Grant.

When a user is a member of more than one group, the blank setting allows conflicts that might occur to be resolved based on any explicitly set assignment.

When a user is a member of more than one usergroup, any explicitly set Grant or Deny overrides a blank.

In the case of conflicting explicitly set assignments, a Deny capability overrides a Grant.

TM1 Application Web capabilities

The Writeback Mode and Sandbox capabilities are ignored by TM1 Application Web.

In TM1 Application Web, the capabilities are set by application. All users associated with a particular application, work in the way set for that application.

If you are working with multiple sandboxes, you must submit from TM1 Application Web, you cannot submit from the Workflow page.

Persistent feeders

To improve reload time of cubes with feeders, set the `PersistentFeeders` configuration parameter to true (T) to store the calculated feeders to a `.feeders` file. Any installation with server load times of over 5 minutes can improve its performance using this parameter.

When this parameter is set to T and the server encounters a persistent feeder file, it loads the saved feeders which reduces the time normally taken to recalculate those feeders. Feeders are saved when the data is saved or rules are edited. You do not explicitly save the feeders

For installations with many complex feeder calculations persisting feeders and then re-loading them at server startup will improve performance. For simple feeders, the time taken to read feeders from disk may exceed the time to re-calculate the feeders but most installations will benefit.

Using the Persistent Feeders feature will increase your system size on disk only. Memory size is not affected by the use of this parameter.

For more information, see also the [CubeSaveData](#) TurboIntegrator function in *TM1 Reference*.

Saving data with persistent feeders

When `PersistentFeeders=T`, and a cube that uses rules is saved, the feeders are stored along side the cube data in a `.feeders` file.

The cube files are called `cube-name.cub` and `cube-name.feeders`.

When a cube is loaded, if a `.feeders` file exists, it is read to re-load the feeders and normal feeder evaluation is skipped. If the feeders file does not exist normal feeder calculation occurs and the `.feeders` file is written after all the feeders have been calculated. This will be the behavior the first time the TM1 server is brought up on a data directory with the `PersistentFeeders=T`. As part of the server startup the `.feeders` files for all the relevant cubes will be written. A new persistent feeders files is written for any cube for which the persistent feeders file is not found at server startup. This means that the `.feeders` files may be deleted from the data directory to force a complete re-calculation of feeders when the server next starts. In a TI process, use the `DeleteAllPersistentFeeders()` TI function to delete all persistent feeders.

The modified time of the `.feeders` file is compared to that of the base cube file at load time. If the cube file is newer than the feeders file, the `.feeders` file is regarded as invalid and is deleted. The `.feeders` file is always written after the cube file so it should always be newer than the cube file.

Modifying rules for cubes with feeders

When rules for a cube are modified the feeders for the rules associated with that cube are re-run to pick up any changes in the feeder statements in the rules file.

The TM1 server never deletes feeders themselves in memory so this recalculation will only add new feeders, it will not delete feeders which may no longer be valid due to the rule file edits.

After the feeders are recalculated, the persistent feeder files for the cube whose rule was edited, and all dependent cubes (due to `DB(...)` statements in the rules) are rewritten. Because feeders in memory are never deleted, this means that rule file edits may leave some feeders set which are no longer valid (no rules actually apply). These unused feeders will be saved with the persistent feeders. Because of the possibility of unused feeders, the administrator may want to periodically (during some time when there is little demand on the server) re-calculate all the feeders. This is done by calling the `DeleteAllPersistentFeeders()` TurboIntegrator function, shutting down the server, and then bringing the server back up. When the server is brought back up, because there is no persistent feeder information, all the feeders will be re-calculated and new persistent feeder information will be written to disk. See the TurboIntegrator Functions chapter of the *TM1 Reference* documentation.

Handling corrupted or invalidated feeder files

If a corrupted or invalid feeder file is detected, all the saved feeders will be deleted and the server will exit with appropriate messages in the log file.

In that case, you must restart the server to reload and re-compute all of the feeders. The messages in the log will be of the form:

```
27660 ERROR 2010-05-14 19:10:26.455 Cognos® .Server All persistent feeder files will be deleted and server will shut down
```

```
27660 ERROR 2010-05-14 19:10:26.455 TM1 .Server Restart server to re-load database.
```

Feeder files are deemed "invalid" if there is a format error in the file, or if the date-time of the feeder file is older than the corresponding cube (.CUB) file which may happen if one tries to manually copy feeder or cube files from one data directory to another. When a feeder file is deemed to be invalid, a message is written to the system log file explaining the exact reason that the feeder file is deemed to be invalid. The message has the form:

```
ERROR 2010-05-14 19:00:22.987 TM1 .Server Invalid feeder file detected for cube "xxxx" reason: ...
```

This message will typically (in the case of a multi-threaded load or an error with an cube which loads early in the startup process) not be near the end of the file, but rather will be in the middle of the load sequence. Searching the message log for "ERROR" will show the error.

Sandboxes and personal workspaces

As the TM1 administrator, you should understand some technical items when using Sandboxes and Personal Workspaces in your TM1 environment.

- Memory usage for Sandboxes and Personal Workspaces
- Managing the files and folders that support Sandboxes and Personal Workspaces
- TM1 Server configuration parameters for Sandboxes and Personal Workspaces

The sandbox feature lets you create your own Personal Workspaces or Sandboxes separate from your base data. A sandbox is not a copy of the base data, but is a separate overlay or layer of your own data values that you have entered on top of the base data.

The new data values that you enter in a Sandbox or Personal Workspace are not saved to the base data until you explicitly commit the changed data in the sandbox to the base data. This behavior is different than working directly in base data where any new data values that you enter are written directly back to the TM1 Server.

Memory usage for sandboxes and personal workspaces

If you are using the sandbox feature, you should be aware that sandboxes require usage of some additional amount of RAM memory and hard disk space. Error messages display when users approach their sandbox memory limit. You can control the amount of memory used for sandboxes per user by adjusting the **MaximumUserSandboxSize** parameter in the TM1 server configuration file, `tm1s.cfg`.

Managing files and folders for sandboxes

Each TM1 server stores the support files for sandboxes in the sub-folders of the TM1 server data directory. If you are backing up this data, you may want to consider these files and the implications of backing up and restoring sandbox files.

Administrators can delete sandboxes using a variety of criteria.

See the `ServerSandboxesDelete` configuration parameter in the *Planning Analytics Installation and Configuration* documentation for more information.

TM1 Server configuration parameters for sandboxes

The TM1 server configuration file, `tm1s.cfg`, uses parameters for enabling and managing the sandbox feature.

- The **DisableSandboxing** parameter specifies whether users have the ability to use sandboxes across the server.

- The **MaximumUserSandboxSize** parameter sets the maximum amount of RAM memory (in MB) to be allocated per user for personal workspaces or sandboxes.
- The use of Personal Workspaces or Sandboxes is controlled using the [Capability Assignments](#) option available on the **Server Explorer > Server menu**.

Understanding sandbox differences among different TM1 clients

The sandbox feature works slightly differently in Cognos TM1 Application Web compared to the legacy clients of TM1 Perspectives, Server Explorer, and TM1 Web.

This information is targeted to those users or administrators who use a combination of these clients.

Default sandbox and base data

Regardless of other configuration settings, in the Cognos TM1 Application Web client, the default sandbox is called default.

This option operates in the same way as the Sandbox feature in other clients. You cannot turn on a Personal Workspace in the Cognos TM1 Application Web client. Cognos TM1 Application Web also does not permit direct writeback so you cannot work directly with the base data when using Cognos TM1 Application Web.

Cognos TM1 Application Web users can submit their sandbox data to base data only as part of the Cognos TM1 Application Web workflow process.

Visibility of sandboxes in different TM1 clients

By design, Cognos TM1 Application Web does not display any sandboxes created in TM1 Perspectives, Server Explorer or TM1 Web.

For example, a sandbox created in TM1 Web does not display in the Cognos TM1 Application Web client sandbox drop-down list.

Understanding Recalc and Commit in a personal workspace

The following table shows the different effects of Commit and Recalc in a Personal Workspace with and without queuing.

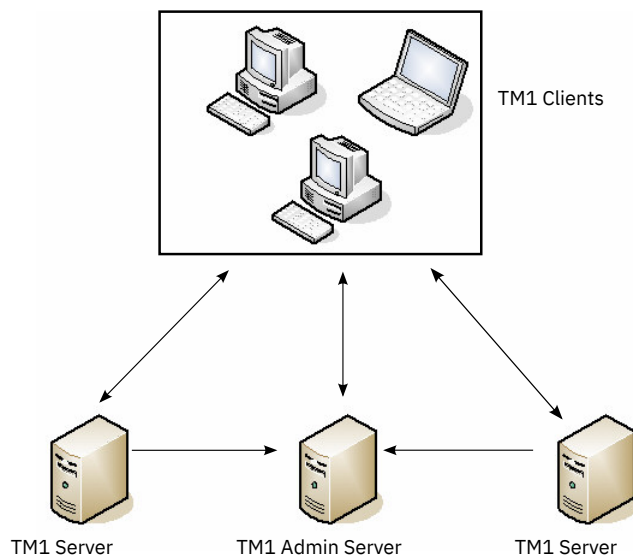
Writeback	Action	Sends Data
Direct	Read--> Recalc Write--> Recalc	From the Base TM1 Model. To the Base TM1 Model.
Personal Workspace	Read--> Recalc Write--> Recalc	From the Base TM1 Model. To the <i>Personal Workspace</i> .
Personal Workspace	Commit	Sends the <i>Personal Workspace</i> data to the TM1 Base Model.
Personal Workspace with Queuing	Commit	Sends the Personal Workspace to the <i>Queue</i> When all resources are available, the Personal Workspace data is processed and sent to the Base TM1 Model.

Chapter 2. TM1 Admin Server operations

This section provides an overview of the IBM TM1 Admin Server and describes the typical administrator tasks of running the Admin Server on Windows and UNIX systems. The TM1 Admin Server is a process that keeps track of all TM1 servers running on a network. An Admin Server runs on a computer known as an Admin Host.

When the TM1 server starts, the server registers itself with an Admin Server that is running on a specified Admin Host. TM1 clients reference the Admin Server to determine which TM1 servers are available on the network.

The following diagram shows how clients and servers use the Admin Server.



The Admin Server maintains the following information for each available TM1 server:

- Server name
- IP address
- Protocol
- Port number

All this information is supplied by the TM1 server when the server registers itself on the Admin Server.

An Admin Server must be running before a TM1 server can start. If you have specified an Admin Host in the `Tm1s.cfg` file or the server command line, the TM1 server will attempt to connect to an Admin Server on that host. The TM1 server will fail to come up if it is unable to connect to the Admin Server for any reason.

If you have not specified an Admin Host, the TM1 server attempts to connect to an Admin Server on the local machine. If an Admin Server is not currently running on the local machine, the TM1 server starts a new Admin Server and connects to it.

The Admin Server becomes aware of TM1 servers on the network by listening for notification from the servers. Usually, the TM1 server sends notification of its presence at a regular interval called the "heartbeat interval," which is 60 seconds by default. When the Admin Server detects the TM1 server, that server becomes registered and available to clients on the network. However, if the Admin Server does not detect the presence of a registered TM1 server over a period equal to three times the heartbeat interval, that TM1 server is removed from the list of servers available on the network. Consequently, the TM1 server will not be available to clients on the network.

By default, the Admin Server uses port 5495. If port 5495 is already in use, you can assign a new port number by creating a new service called Tm1admsrv. All TM1 applications look for a named service called Tm1admsrv, and if that service exists, the applications use the port number assigned to the service. If the service does not exist, TM1 applications use port 5495.

Running the TM1 Admin Server on Windows

After you install the IBM TM1 Admin Server on a Microsoft Windows system, you can manage the server either manually or by using the Cognos Configuration utility.

You can also run specific TurboIntegrator processes as part of startup.

See the "Chores" chapter in the *TM1 TurboIntegrator* documentation for more information.

Manually starting the TM1 Admin Server on Windows

You can manually run the TM1 Admin Server as a service or an application on Microsoft Windows.

Use the Tm1admsd.exe file in the *install_dir\bin* directory to run the Admin Server as a Windows service.

- To add the TM1 Admin Server as a Windows service, run the following command at a command prompt from the *install_dir\bin* directory:

```
Tm1admsd -install
```

- To remove the TM1 Admin Server Windows service, run the following command at a command prompt from the *install_dir\bin* directory:

```
Tm1admsd -remove
```

Use the Tm1admsrv.exe file to run the Admin Server as a Windows application. For details, see [“Running the TM1 Admin Server as an application on Windows”](#) on page 10.

Using Cognos Configuration to manage the TM1 Admin Server on Windows

For details on using the Cognos Configuration utility to manage the TM1 Admin Server on a Microsoft Windows operating system, see *Planning Analytics Local Installation and Configuration*.

Running the TM1 Admin Server as an application on Windows

To run the Admin Server as an application, use the Tm1admsrv.exe executable file. This file is placed in the *install_dir\bin* directory when you install IBM Planning Analytics. You can run the Admin Server by double-clicking the tm1admsrv.exe file, but we recommend that you create a shortcut to the executable file in the startup directory on the Admin Host. Then, the Admin Server is launched whenever the Admin Host starts.

Procedure

1. Click the **Start** button on the Windows taskbar.
2. Choose **Settings, Taskbar**.
3. Click the **Start Menu Programs** tab.
4. Click **Add**.

The Create Shortcut dialog box opens.

5. Enter the full path to Tm1admsrv.exe in the **Command line** field, or click **Browse** to navigate to the file.
6. If necessary, specify a command line parameter to set the heartbeat interval for the Admin Server.

Parameter

- h

Sets the heartbeat interval, in seconds, for TM1 Servers registered with the Admin Server.

The default heartbeat interval is 60 seconds.

7. Click **Next**.

The Select Program Folder dialog box opens.

8. Select **Startup**.

9. Click **Next**.

The Select a Title dialog box opens.

10. Type a name for the shortcut and click **Finish**.

Viewing Admin Server Status

When an Admin Server is running as an application on an Admin Host, an icon is appended to the Windows system tray.

To view the current status of the Admin Server, double-click the icon.

The IBM TM1 Admin Server window shows the following information:

- Time at which the Admin Server was started
- Machine on which it is running
- Port being used
- Details of each TM1 Server currently registered with the Admin Server

To shut down an Admin Server running as an application, click **Stop Admin Server**.

You cannot view the status of an Admin Server running as a Windows service.

Running the TM1 Admin Server on UNIX

After you install the TM1 Admin Server on a UNIX system, you can start, stop and manage the Admin Server either manually or by using the IBM Cognos Configuration utility.

Manually starting the TM1 Admin Server on UNIX

To manually run the TM1 Admin Server on a UNIX system, run the `tm1admsrv.exe` file from the `install_dir/bin` directory.

Using Cognos Configuration to manage the TM1 Admin Server on UNIX

For details on using the IBM Cognos Configuration utility to manage the TM1 Admin Server on a UNIX system, see *Planning Analytics Installation and Configuration*.

Viewing Admin Server Status Report

To view an Admin Server status report, run `tm1admstat.exe` from the `install_dir/bin` directory.

The Admin Server report contains the following information:

- Time at which the Admin Server was started
- Machine on which it is running
- Port being used
- Details of each TM1 Server currently registered with the Admin Server

Specifying the location of the Admin Host

You specify the location of the Admin Host differently for clients (user interfaces) and remote servers.

Specify the Admin Host for clients

- To specify the Admin Host referenced by clients, edit the AdminHost parameter in the Tm1p.ini client configuration file.

You can change the Tm1p.ini file by using the TM1 Options menu in Server Explorer.

You can also manually edit the Tm1p.ini file. For more information, see the Tm1p.ini topic in *Planning Analytics Local Installation and Configuration*.

Specify the Admin Host for remote servers

- To specify the Admin Host with which remote servers register, use one of the following methods:
 - Edit the AdminHost parameter in the Tm1s.cfg file.
 - Use the -v command-line parameter when you bring up the Windows version of the TM1 Server.

For information on server configuration parameters, see the topic about the Tm1s.cfg file in *Planning Analytics Local Installation and Configuration*.

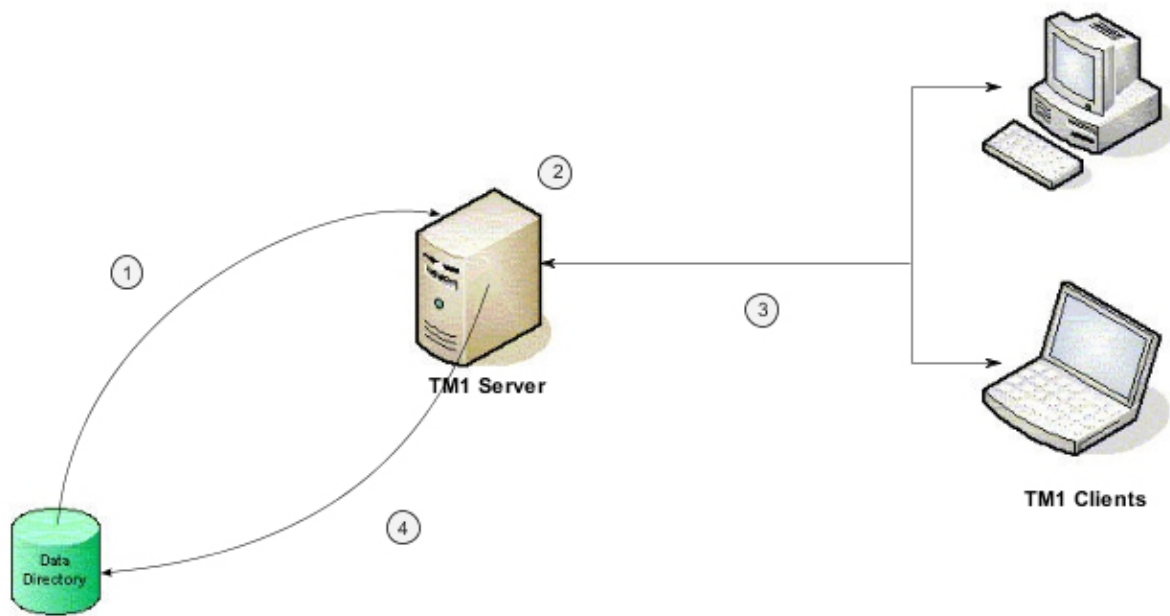
Specifying multiple Admin Hosts

You can set a TM1 client to reference multiple Admin Hosts by separating host names with semicolons.

A client that specifies multiple Admin Hosts can access any TM1 Servers that are registered with the Admin Servers on the specified hosts.

Chapter 3. Remote TM1 Server operations

This section describes how IBM TM1 remote servers work, and how you can manage their operations. The following diagram illustrates the operations of a remote TM1 Server.



1. When the TM1 server is started, all TM1 data is loaded from the TM1 data directory into RAM on the server machine.
2. While the TM1 server is running, all cube data resides in RAM. All edits received from TM1 clients are stored in a transaction log file named Tm1s.log.
3. TM1 clients retrieve cube values from the server. Clients also send edits to cube values to the TM1 server.
4. When the TM1 server is shut down, or when an explicit **Save Data** command is issued, any changes to cube values are written from the transactional log file to the data directory.

For more information, see [TM1 Server overview](#).

Setting up a remote TM1 Server to run as an application

After you run the Planning Analytics installation wizard on a Microsoft Windows system and install the sample TM1 databases, you can use the IBM Cognos Configuration utility to run each sample database as a Microsoft Windows service. Or, you can manually set up the TM1 Server to run as an application.

Procedure

1. Open the *pa_install_directory*\bin directory in Windows Internet Explorer.
2. Right-click Tm1s.exe.
3. Select **Create Shortcut**.
Windows creates Shortcut to Tm1s.exe in the *pa_install_directory*\bin directory.
4. Right-click Shortcut to Tm1s.exe.

5. Select **Properties**.

The Properties window opens.

6. Click the **Shortcut** tab.

The Target field on this tab contains the full path to Tm1s.exe.

7. In the Target field, add the -z flag and specify the full path to the directory containing the Tm1s.cfg file for the server.

For example, -z C:\TM1\salesdata indicates that the Tm1s.cfg file for the server resides in the C:\TM1\salesdata directory.

Note: If the path to the directory contains any blank spaces, enclose the entire path with double quotes.

The contents of the Target field should resemble the following:

```
"C:\Program Files\IBM\cognos\tm1\bin\tm1s.exe" -z C:\TM1\salesdata
```

8. Click **OK** to save the shortcut.

You might want to move the shortcut to the desktop for easy access.

Starting a TM1 Server set up as a Windows application

You can start a remote server that has been set up as a Windows application.

Procedure

Double-click the TM1 Sales Data icon.

Results

A server window opens. The server window lists the cubes and dimensions loaded into the server RAM.

Shutting down a TM1 Server set up as a Windows application

You can shut down a remote server that has been set up to run as a Windows application.

Procedure

Click the **Close** button in the server window.

Setting up a TM1 Server to run as a Windows service

You can manually install a TM1 Server to run as a Windows service, or remove an existing one, using the command line options of the tm1sd.exe file. The tm1sd.exe file is installed in the *pa_install_directory*\bin directory.

To install a TM1 server to run as a Windows service, use the following command line format: `tm1sd.exe -install parameters`. The following table lists the available parameters.

Parameter	Required/Optional Description
-n	Required Name of the TM1 Server. This will become the Windows service name as shown when you display the properties of an installed service.
-z	Required Data directory that contains the tm1s.cfg file for the TM1 database.

Parameter	Required/Optional Description
-u	Optional Valid user name for the computer on which you are working. Use one of the following formats: <ul style="list-style-type: none"> • Domain and user name in the format <i>Domain\username</i>. For example, entp\jsmith. • User name for a local user account. If you do not provide a user name, the current user account is used by default.
-w	Optional Password for the above account.

For example, the following command line installs the sdata TM1 server as a Windows service for the current user.

```
tm1sd.exe -install -n sdata -z C:\Program Files\Cognos\TM1\Custom\TM1Data\SData
```

Starting a TM1 Server installed as a Windows service

To start a remote server that has been installed as a service, complete the following steps.

Procedure

1. Open the Microsoft Windows Services manager.

A list of installed services opens.

2. Select the entry for the TM1 server you want to start, for example **Planning Sample**, and click the **Start** button.

An Admin Server must be running before a TM1 server can start. If you have specified an Admin Host in the Tm1s.cfg file, the TM1 server will attempt to connect to an Admin Server on that host. The TM1 server will fail to come up if it is unable to connect to an Admin Server for any reason.

If you have not specified an Admin Host, the TM1 server attempts to connect to an Admin Server on the local machine. If an Admin Server is not currently running on the local machine, either as a service or an application, the TM1 server starts a new Admin Server application and connects to it.

Setting up a service to start automatically

To set up an installed service to start automatically when Microsoft Windows reboots, complete the following steps.

Procedure

1. Open Windows Services manager.

A list of installed services opens.

2. Double-click the **TM1 Server** entry.

The **Service Properties** dialog box opens.

3. Select **Automatic** as the Startup Type and click **OK**.

Removing a TM1 Server from running as a Windows service

To remove a TM1 Server that is running as a service, use the following command line format.

```
tm1sd.exe -remove -n <ServerName>
```

where <ServerName> is the Microsoft Windows service name of the TM1 server you want to remove. For example, the following command line removes an existing service for the sdata TM1 Server.

```
tm1sd.exe -remove -n sdata
```

Shutting down a TM1 Server running as a Windows service

You can shut down a remote server that has been installed as a Microsoft Windows service.

Procedure

1. Open the Microsoft Windows Services manager.

A list of services opens.

2. Right-click the **TM1 Server** and select **Stop**.

For details and additional steps on shutting down a server from a remote location when logged in as the TM1 Admin user, see [“Managing client connections” on page 20](#).

Results

The TM1 Server running as a Windows service automatically shuts down during system shutdown. In some instances, a server running a large or complex model, might not properly shut down during system shutdown, power failure, or manual service shutdown. An improper shutdown is defined as a shutdown in which the TM1 Server is summarily terminated before it has completed all shutdown procedures. This can happen during normal system shutdown because Windows allots a limited time (approximately 20 seconds) for service shutdown. After the allotted time expires, the system shutdown proceeds regardless of whether the service shutdown is complete.

When the TM1 Server is running as a service, it automatically recovers any data changes from the previous sessions in which an improper shutdown occurs. The changes are recovered from records in the server's Tm1s.log file.

For details on Microsoft's approach to service shutdown or on increasing the amount of time allotted by Windows for service shutdown, see the Microsoft support website.

Starting a UNIX TM1 Server

To assist you in starting a UNIX TM1 Server, a sample script named startup_tm1s.sh is available in the *pa_install_directory/bin* or */bin64* directory.

With a few minor modifications, you can use this sample script to start your UNIX TM1 Server. When you use the script to start a server, TM1 uses the Tm1s.cfg file in the specified data directory to direct the behavior of the server.

Modifying the startup_tm1s.sh script

Modify the path parameters to customize the startup_tm1s.sh script.

Procedure

1. Open the file in a text editor.
2. Set the tm1_path parameter to the directory containing the tm1s.exe server executable file. In most circumstances, this directory is *pa_install_directory/bin*.
3. Set the tm1_data_path parameter to your server data directory.
4. Set the TM1_PATH parameter to directory containing the tm1s.exe server executable file. In most circumstances, this directory is *pa_install_directory/bin*.

Note: You must set both tm1_path and TM1_PATH to the same directory.

5. Save the script.

For example:

```
> ./startup_tm1s.sh /software/AutoInstall/TM1/samples/tm1/SData
```

You can save to a new file name if desired. If you have installed multiple TM1 servers, you should create server-specific scripts with unique file names for each server.

6. Run ./startup_tm1s.sh (or the new file name) to start your TM1 server.

Running the UNIX TM1 Server in background mode

You can run the TM1 Server in background mode by adding the parameter `RunningInBackground` to the `tm1s.cfg` file and setting the parameter to `T`.

```
RunningInBackground=T
```

You must manually add the `RunningInBackground` parameter to `tm1s.cfg`. This parameter is not part of the standard `tm1s.cfg` file created when you install the TM1 Server. For details about server configuration parameters, see *Planning Analytics Installation and Configuration*.

See also [“Shutting down a UNIX TM1 Server running in background mode”](#) on page 17.

Shutting down a UNIX TM1 Server

You can shut down a server running in the foreground.

Procedure

1. Press CTRL+C in the window from which you started the server.
2. Before you exit, the server prompts you to save all data to disk. Press ENTER (or RETURN) to save all data. Type No and press ENTER to shut down the server without saving the data.

Shutting down a UNIX TM1 Server running in background mode

To shut down the TM1 Server running in background mode, run the shutdown utility, `tm1srvstop.exe`, located in the `pa_install_directory/bin` directory.

The shutdown utility `tm1srvstop.exe` accepts the following parameters:

Parameter	Description
-n	Required The name of the TM1 Server you want to shut down. Note that this server must have been started in background mode.
-v	Required IP address of the Admin Server on which the TM1 Server is registered.
-user	Required User name of the person bringing down the server. This person must have the necessary access and privileges to shut down the server.
-pwd	Required Password for the person bringing down the server.

Parameter	Description
-time	Optional Delay, in seconds, at which the server will be shut down after running shutdown_tm1s.sh
-cancel	Optional Cancels any previously issued shutdown_tm1s.sh commands. Valid values are T and F.
-unsave	Optional By default, TM1 saves the data before a TM1 Server is shut down with the shutdown_tm1s.sh command. The -unsave parameter shuts down a server without saving the data. Valid values are T and F.

For example, the following command line allows a user named Admin with a password of Administrator to shut down a TM1 Server named TM1SERV registered on an Admin Host located at 207.110.100.206. The command line indicates a 30-second interval between the time the command is issued and the time the server shuts down. The server is shut down without saving the data.

```
tm1srvstop.exe -n TM1SERV -v 207.110.100.206 -user ADMIN
-pwd Administrator -time 30 -unsave T
```

Alternatively, you can use shutdown_tm1s.sh to shut down a server. This method has the following limitations, however:

- The shutdown_tm1s.sh script works only on the local machine where TM1 Server is running.
- If you use the startup_tm1s.sh and shutdown_tm1s.sh scripts, you must ensure that the following line exists in the TM1 Server configuration file (tm1s.cfg): RunningInBackground=T .

Connecting to a remote server

To connect to a remote TM1 server, a client must point to an Admin Host on which an Admin Server is running. The Admin Server maintains information about remote TM1 Servers available on the network, including name, protocol, address, and port number. The Admin Host used by the client is specified in the Tm1p.ini file.

Procedure

1. Access Server Explorer.
2. Double-click the icon for the server to which you want to connect.

The Server Login dialog box opens.

3. Enter the correct user name and password, and click **OK**.

If the connection to the server is successful, TM1 displays the Cubes, Dimensions, Replications, Processes, and Chores icons for that server. (The Replications icon displays only if you logged in as a member of the ADMIN group.)

Note: The TM1 remote server comes with a predefined administrator ID of admin and a password of apple.

Refreshing the list of remote servers

If you do not see the remote server to which you want to connect in the list of available servers, you can refresh this list.

Procedure

Click **File > Refresh Available Servers**. TM1 displays all running servers that are registered on the Admin Host to which the client is pointing.

Resetting the Admin Host

You can set or change the Admin Host.

Procedure

1. From Server Explorer, click **File > Options**.

The TM1 Options dialog box opens.

2. Enter the name of the Admin Host in the Admin Host field. You can concatenate two or more hosts by separating the host names with a semicolon.

You can also click the drop-down menu to select from previously accessed Admin Hosts.

3. Click **OK**.

When you change the Admin Host, TM1 disconnects you from the remote servers to which you are connected and restarts your TM1 client. If a local server is running, that server is shut down and restarted.

Resetting local server options

The following table describes the local server options that you can change through the TM1 Options dialog box.

Local Server Option	Description
Data Directory	Directory for the local server data files.
Connect to Local Server on Startup	Select or clear the Connect to Local Server on Startup check box to control whether the TM1 client automatically connects to the local server at startup.

Disconnecting from a remote server

You can disconnect from a remote server.

Procedure

1. In Server Explorer, select the icon for the server that you want to disconnect from.
2. Click **Server > Log Out**.

Logging transactions

Each TM1 server tracks the data transactions made by its clients. When a client changes a cube value, TM1 records the change in a transaction log file named Tm1s.log, which is located in the TM1 server data directory.

For more information, see [“The transactions log” on page 44](#) in [“Planning Analytics and TM1 logs” on page 37](#).

Monitoring server performance

TM1 includes a performance monitoring feature that lets you record the performance statistics for clients, cubes, and servers. When you enable performance monitoring, TM1 populates several control cubes on a minute-by-minute basis. You can then browse these cubes to analyze the server performance.

For more information, see [“Control cubes” on page 59](#) and [Appendix B, “Control Cubes,” on page 121](#).

Managing client connections

As a TM1 Server administrator, you can use the Clients Messaging Center to manage client connections.

Tasks include:

- Broadcasting Messages to Clients
- Disconnecting Clients from the Server
- Shutting Down a Server Remotely

Broadcasting messages to clients

If you are a member of the ADMIN group for a TM1 Server, you can broadcast messages to any clients that are connected to the server. For example, you can inform clients that they are going to be disconnected from the server or that the server is going to be shut down.

Procedure

1. In the left pane of Server Explorer, select the server on which you want to broadcast a message.
2. Click **Server, Server Manager**.

The Clients Messaging Center dialog box opens.

3. Select **Do nothing** to broadcast the message without shutting down the TM1 Server.
4. Select **Broadcast a Message**.
5. Enter the message you want to broadcast in the Broadcast Message box.
6. Click **Select Clients**.

The Subset Editor opens with a subset of all the clients that are currently connected to the TM1 Server. You cannot send messages to clients that are not currently connected to the server.

7. Select the clients you want to receive the message and click **OK**.
8. Click **OK** in the Clients Messaging Center to broadcast the message.

Disconnecting clients from the server

You can disconnect clients from a TM1 Server.

Procedure

1. In the left pane of Server Explorer, select the server from which you want to disconnect clients.
2. Click **Server, Server Manager**.

The Clients Messaging Center dialog box opens.

3. Select **Disconnect Clients**.
4. Specify a Minutes interval to determine when the clients will be disconnected.
5. Click **Select Clients**.

The Subset Editor opens with a subset of all clients that exist on the server. The subset is not a subset of all clients currently connected.

6. Select the clients you want to disconnect and click **OK**.

A message is sent to all selected clients with a warning that they will be disconnected at the interval you specified in Step 4.

7. If you want to broadcast a more detailed message to the selected clients, you can select the Broadcast Message to Selected Clients option and enter a message in the Broadcast Message box.
8. Click **OK** in the Clients Messaging Center.

Shutting down a server remotely

You can shut down a server remotely from a TM1 client.

Procedure

1. In the left pane of Server Explorer, select the server you want to shut down.
2. Click **Server > Server Manager**.
The Clients Messaging Center dialog box opens.
3. Select **Shutdown Server**.
4. Specify a Minutes interval to determine when the server will be shut down.
5. Click **OK**.

Any clients currently connected to the server receive a message that the server will be shut down in the number of minutes you specified in Step 4.

Remote server memory management

The TM1 Server uses a sparse memory management scheme, which allows the server to hold very large cubes in much less space than the same data would occupy in a relational database. Therefore, the memory management scheme allows the server to accommodate very large databases in RAM without the need to use disk space.

The RAM that is used by the server is not static. When the remote server calculates the consolidations, the server stores the results for later reference. The second time a consolidated value is requested, TM1 can deliver the value without calculation, which provides a great improvement in speed.

The remote server memory management approach means that the longer the server runs, the faster it becomes, as an ever-increasing number of consolidations are stored and do not need to be recalculated. The amount of memory the server uses increases incrementally. The server does not take memory byte-by-byte, but rather takes a larger piece every so often.

TM1 does not release the memory back to the operating system until the TM1 Server is terminated. Instead, TM1 puts memory it no longer needs into a garbage list, to be reused as required. Accordingly, the memory consumption for the TM1 Server, as reported by the operating system, is the total of the actual current memory usage and the garbage memory. You can view an accurate report of the actual memory usage and garbage memory by viewing the `StatsForServer` control cube, which is described in Appendix B, “Control Cubes,” on page 121.

All the remote server platforms support virtual memory, where disk space is used as if it were RAM. The server runs in virtual memory, but it slows down the performance significantly. The best practice is for you to keep all the cubes in real RAM at all times.

You should keep an eye on the memory consumption by using the system-monitoring utilities. If the system runs slowly, you might need to add more RAM to your server.

Some operating systems maintain a configurable limit on the memory they will allocate to any one process. If your system appears unable to take all the memory that should be available to it, you might want to look at the operating system parameters.

You can limit the amount of memory allocated to any individual view by adding the `MaximumViewSize` parameter to the `Tm1s.cfg` server configuration file. For more details about configuration parameters, see *Planning Analytics Installation and Configuration*.

About Stargate views

A Stargate view is a calculated and stored subsection of a TM1 cube that TM1 creates when you browse a cube with the Cube Viewer or In-Spreadsheet Browser. The purpose of a Stargate view is to allow quicker access to the cube data.

A Stargate view is different from a TM1 view object. The Stargate view contains only the data for a defined section of a cube, and does not contain the formatting information and browser settings that are in a view object.

A Stargate view that TM1 creates when you browse a cube in the Cube Viewer or In-Spreadsheet Browser contains only the data defined by the current title elements and row and column subsets.

TM1 stores a Stargate view when you access a view that takes longer to retrieve than the threshold defined by the VMT property in the }CubeProperties control cube. (If a VMT value is not explicitly defined, a Stargate view is generated when a view takes longer than five seconds. This is the default threshold when VMT is not specified in the }CubeProperties control cube.)

A Stargate view persists in memory only as long as the browser view from which it originates remains unchanged. When you recalculate the browser view, TM1 creates a new Stargate view based on the recalculated view and replaces the existing Stargate view in memory. When you close the browser view, TM1 removes the Stargate view from memory.

Using TM1 in bulk load mode

Bulk Load Mode enables IBM TM1 to run in a special optimized single-user or single chore/process mode. This mode can maximize TM1 performance for dedicated tasks during times when little or no other activity is expected.

Some examples of using Bulk Load Mode include:

- TM1 administrators that needs to manually perform maintenance operations.
- A night-time window to load large amounts of data.

TM1 typically runs in a multi-user mode where multiple users, chores and processes can all run concurrently accessing TM1 data. In Bulk Load Mode, the TM1 server prevents concurrent activity by temporarily suspending other users, chores and processes and eliminates the overhead required by a multi-user environment.

Bulk Load Mode doesn't actually log out users, but only suspends their interaction with TM1. As soon as Bulk Load Mode is done, any users that were previously logged in are reactivated and user-interaction with TM1 resumes.

You can enable Bulk Load Mode directly within a TI process or with the TM1 API. In either case, you use commands to enter and leave Bulk Load Mode.

Considerations for using bulk load mode

You should consider the following when using bulk load mode.

- Bulk load mode does not display a message to end-users to alert them. You will need to plan and coordinate your usage of Bulk Load Mode accordingly.
- Only a single user or process may be active during Bulk Load Mode. No new connections can be established to the server while it is operating in Bulk Load Mode.
- A TI process can not use the ExecuteCommand to launch a command line program that attempts to log back into the same TM1 server. The login attempt will fail.
- Any scheduled chores that are scheduled to execute during the time Bulk Load Mode is enabled are deactivated and not run.

Starting bulk load mode

When the server enters bulk load mode, all processing by other threads is paused.

Any existing user threads and running chores will be suspended. Only the thread that initiated bulk load mode will remain active. All scheduled chores will be deactivated, except the chore that initiates bulk load mode. All system-specific threads connections will also be suspended.

Ending bulk load mode

When bulk load mode is disabled, all system and user threads will be resumed and user logins will be allowed.

Custom applications that use the TM1 API to enable bulk load mode should also call the necessary TM1 API function to *exit* bulk load mode. However, if the client connection is severed (the network fails or the client logs out, crashes or disconnects), the server will automatically exit bulk load mode.

Similarly, if a TI process/chore is running in bulk load mode and the process exits, whether successfully or with errors, the server will automatically exit bulk load mode.

When the server returns to normal multi-user mode, any chores that were deactivated get reactivated and return to their normal schedule. If the chores were scheduled to run, but were prevented by bulk load mode, they will not get executed immediately, but will execute according to the schedule. You may have to adjust the launch time of your scheduled chores to prevent them from getting locked out during the times you enable bulk load mode.

TM1 C API

Use the following TM1 API functions to enable and disable Bulk Load Mode.

```
TM1V TM1ServerEnableBulkLoadMode(TM1P hPool, TM1Server hServer)
```

```
TM1V TM1ServerDisableBulkLoadMode(TM1P hPool, TM1Server hServer)
```

where:

- *TM1P* is a pool handle obtained with `TM1ValPoolCreate`.
- *TM1Server* is a handle to the current server.

TurboIntegrator process commands

You can enable bulk load mode in either the Prolog or Epilog section of a TurboIntegrator process. For efficiency, we recommend enabling bulk load mode in the first, or very close to the first, statement in the Prolog section of your process.

After enabling bulk load mode in a process, it can only be disabled on the last line in the Epilog section. If you attempt to disable bulk load mode anywhere else in the process, the process will not compile.

If the mode is enabled in one TurboIntegrator process, it remains enabled until explicitly disabled or until the chore completes. This means you can enable the mode in a process within a chore and then run a series of TurboIntegrator processes before disabling it. You can also enter and exit bulk load mode repeatedly, using the mode only for certain critical parts of a chore.

Use the `EnableBulkLoadMode()` and `DisableBulkLoadMode()` TurboIntegrator commands to enable and disable bulk load mode in a TurboIntegrator process.

The `DisableBulkLoadMode()` function can be used only on the last line in the Epilog section of your TurboIntegrator process when using bulk load mode.

Chapter 4. Planning Analytics security overview

You can control access and authentication to the servers.

The security features in IBM Planning Analytics enable you to control authentication and object security. This defines who logs in to your IBM TM1 Server and which objects those users are allowed to access.

For complete details about configuring authentication security, see *Planning Analytics Local Installation and Configuration* documentation.

Overview to authentication

You can configure IBM TM1 Server to use a specific authentication mode to control user login access.

By default, when you install the TM1 Server it is configured to use the standard Cognos TM1 authentication.

After you install the TM1 Server, you can change the authentication method by changing the parameters in the TM1 configuration files.

Authentication Methods	Description
TM1 Authentication	TM1 Server asks you for a user name and password, and validates the login information against the security cube login information.
Integrated Login	Microsoft Windows performs the Cognos TM1 authentication.
LDAP Authentication	TM1 Server asks you for a user name and password, and validates the login information against an external LDAP server.
IBM Cognos Security	The TM1 Server authenticates users using IBM Cognos security.

TM1 object security

As the Planning Analytics administrator, you can control access to objects by assigning specific levels of object security to groups.

Suppose you have a group named Executives in your TM1 database. You want these executives to review the company budget summary data, which is stored in the BudSummary cube in your TM1 database. You can use TM1 to assign the Executives group Read privileges to the BudSummary cube.

You set TM1 access control the same way for any of the products in the TM1 suite. For example, if users in the Executives group have Read access to the BudSummary cube when they run TM1 Perspectives, they also have Read access to that cube when they run the TM1 Web client.

To learn more about the procedures that you use to set security for TM1 objects, see the *TM1 for Developers* documentation.

Chapter 5. Managing users and groups

In IBM Planning Analytics, security is based on the groups that users belong to.

Users and groups security overview

Planning Analytics manages security by organizing users into groups. Planning Analytics includes a set of predefined administrative groups and also lets you create your own custom groups. Users can belong to one or multiple groups.

TM1 has two major types of user groups:

Administrative Groups

Includes only the predefined groups of ADMIN, DataAdmin, SecurityAdmin, and OperationsAdmin. For more information, see [“Administrative groups and authority” on page 28](#).

User Groups

Includes all user-created groups, which typically contain non-administrative TM1 users.

You can use the TM1 Clients/Groups box to manage TM1 users and groups. The Clients/Groups dialog box organizes groups and users separately.

As a TM1 administrator, you must be running TM1 Architect or TM1 Perspectives to administer security on the IBM TM1 server.

Implementing a TM1 security scheme

You can implement a TM1 security scheme to secure objects on a remote server.

Procedure

1. Define security groups.
2. Assign access permissions to the groups.
3. Define the TM1 users.
4. Assign users to groups.

You can implement the following security levels:

Cube-level security

Controls access to cubes.

Element-level security

Controls access to the cells identified by elements.

Dimension-level security

Controls the ability to add, remove, and reorder the elements in a dimension. This type of security is independent of cube-level and element-level security, except when users have no access to a dimension.

Cell-level security

Controls access to a cell.

Note: A TM1 client and server communicate directly over the network with a TM1 user name and password. This security is independent of standard file system security and is not affected by user permissions on network directories or file servers.

User and group security examples

To illustrate IBM Planning Analytics security and let you experiment with security features, the TM1 sample data (Sdata) that is installed with the Planning Analytics comes with a set of pre-defined groups and users.

To view these groups and users, right-click the Sdata server in the Server Explorer and click **Security > Clients/Groups**.

The TM1 Clients/Groups dialog box appears.

The following table contains the passwords for the sample users.

Note: Passwords are hidden in the Clients/Groups window and display as either Defined or Undefined.

User name	Password	Group Membership
Admin	apple	ADMIN
Usr1	red	North America
Usr2	orange	South America
Usr3	blue	North America and South America

Administrative groups and authority

IBM Planning Analytics supports the separation of administrative duties and roles in Planning Analytics by dividing administrative users into the following predefined administrator groups:

ADMIN group

Members of the ADMIN group have access to all areas of TM1 and represent super-users with all privileges.

SecurityAdmin group

The SecurityAdmin group can perform only security operations in TM1. These operations include creating, editing, and deleting TM1 users and groups. This group can manage the access permissions of other users to TM1 objects, such as cubes, dimensions, and rules, but this group cannot view the data in those same TM1 objects.

DataAdmin group

The DataAdmin group has ADMIN privileges to everything that is not related to security. This group can view, edit, and save TM1 objects, such as cubes, dimensions, rules, and processes. Members in this group can view security settings in read-only mode but are not allowed to modify security settings.

OperationsAdmin group

The OperationsAdmin group can perform only maintenance operations in TM1. This group can perform TM1 server maintenance and operational work, for example, disconnect users or cancel threads. Members in the OperationsAdmin group have no access to any TM1 metadata, cube data, or any other data.

The security assignments for these administrator groups are hardcoded and cannot be modified.

You can use these predefined administrator groups to control and separate TM1 administrative roles among different users to satisfy internal or external security requirements and rules.

Note: Replication and synchronization operations in TM1 should be performed only by members of the ADMIN group. Members of the DataAdmin, SecurityAdmin, and OperationsAdmin groups do not have all the access privileges to perform these operations.

ADMIN group

The ADMIN group, which is created for administrative purposes only, should contain only a small group of trusted TM1 administrators.

Members of the ADMIN group always have the highest level of security for all objects on the remote server, and can do the following administrative tasks:

- Assign and revoke other user's access permissions.
- Create and delete cubes, dimensions, elements, processes, and chores.
- Add, remove, and reorder elements in a dimension.
- Change cube data.
- Create and delete public views and subsets.
- Reserve and lock cubes, elements, and dimensions, so that other users cannot update them.
- Remove reservations and locks from cubes, elements, and dimensions.
- Create replicated objects on the server.
- Change TM1 cell security.
- Create and delete TM1 rules.

A user who is a member of multiple groups receives the highest level of permissions from all groups. For example, joint membership in the ADMIN group and SecurityAdmin group or the ADMIN group and the DataAdmin group is treated as ADMIN. Some restrictions apply for each predefined group. For more information, see the definition of each group.

Members of the ADMIN group are the only TM1 users that can add users to the ADMIN group.

SecurityAdmin group

Members of the SecurityAdmin group can access the security-related features of TM1, but cannot view the data in non-security TM1 objects, such as cubes, dimensions, and rules.

The members in this group are allowed to create, edit, and delete TM1 users and groups. Additionally, these users can manage the access permissions of other users to TM1 objects such as cubes and dimensions.

If integrated login is not being used, the security administrator is also able to reset user passwords.

As a member of the SecurityAdmin group, you can manage TM1 security with the following tools in Server Explorer.

Clients/Groups window

Assign TM1 clients to TM1 groups.

Security Assignments windows

Control user access to TM1 objects such as cubes, dimensions, and processes.

Security control cubes

Manually apply security privileges for TM1 objects and user groups.

Restrictions on replication and synchronization

Members of the SecurityAdmin group do not have all the required access privileges to perform replication and synchronization operations in TM1 and should not attempt to perform these operations.

Restrictions for rules and processes

Members in the SecurityAdmin group cannot write or modify rules and processes. They cannot view rules, but they can view processes in read-only mode.

For more information about processes, see the *TM1 for Developers* documentation.

Combining SecurityAdmin membership with other groups

Membership in the SecurityAdmin group is not intended to be combined with membership in the DataAdmin group or any other user group. The SecurityAdmin is restricted from accessing non-security

objects and these restrictions always apply regardless of what other groups the user belongs to. Additionally, TM1 does not allow users to be added to any other user group after they have been assigned to the SecurityAdmin group.

These restrictions prevent the SecurityAdmin from adding him or herself to another group to gain access to data or operations that are not allowed for the SecurityAdmin.

Using the SecurityAdmin Group with the TM1 C API

The TM1 C API does not allow programmers to configure joint membership with the SecurityAdmin group. The ClientGroupAssign function rejects any attempt to place a user that is a member of the SecurityAdmin group into another group.

Restrictions on adding users to the ADMIN group

Members of the SecurityAdmin group are not allowed to add users to the ADMIN group. Only members in the full ADMIN group can add other users to the ADMIN group. This prevents the SecurityAdmin from creating a user account in the ADMIN group that they could then use with full administrative privileges.

DataAdmin group

Members in the DataAdmin group have ADMIN privileges to everything that is not related to security. Users in this group can view, edit, and save TM1 objects, including cubes, dimensions, rules, and processes.

The DataAdmin group can view security settings in read-only mode, but members of this group are not allowed to modify any security settings.

For example, as a member of the DataAdmin group, you can open the Clients/Groups window or any of the Security Assignments windows for TM1 objects (cubes, dimensions, processes). However, these windows display in read-only mode and you cannot make changes.

Membership in user groups

Combined membership in the DataAdmin group and any other user group is allowed. However, this combination is treated as the DataAdmin group. Any restrictions that are imposed by the user group access are overridden by the DataAdmin access.

For example, if a user is a member of the DataAdmin group and a user group, and that user group does not have security permissions to a cube, the user can see the cube based on their DataAdmin permissions.

Membership in both the SecurityAdmin and DataAdmin groups

Membership in the DataAdmin group is not intended to be combined with membership in the SecurityAdmin group.

If a user belongs to both the DataAdmin group and the SecurityAdmin group, their security permissions are SecurityAdmin only. Being a member in both the SecurityAdmin and DataAdmin group allows the user to view and edit security settings for the users and groups.

Restrictions on replication and synchronization

Members of the DataAdmin group do not have all the required access privileges to perform replication and synchronization operations in TM1 and should not attempt to perform these operations.

Restrictions on rules for security control cubes

A DataAdmin cannot create rules on security control cubes. This restriction prevents a DataAdmin from creating a rule that might modify cells in a security control cube.

OperationsAdmin group

The OperationsAdmin group can perform only maintenance operations in TM1. Members in the OperationsAdmin group have no access to any TM1 metadata, cube data, or any other data.

As a member of the OperationsAdmin group, you can perform the following TM1 Server operations with the TM1 REST API.

View threads

View threads running in the TM1 server.

Example

Method: GET

Syntax:

```
http://localhost:5657/api/v1/Threads
```

Cancel threads

Cancel an operation.

Example

Method: POST

Syntax:

```
http://localhost:5657/api/v1/Threads('13124')/tm1.CancelOperation
```

Disconnect users

Disconnect a user but existing threads that are active continue to run until they have completed. An OperationsAdmin can disconnect a member of the Admin group. This approach is a change from previous behavior.

Example

Method: POST

Syntax:

```
http://localhost:5657/api/v1/Users('Admin')/tm1.Disconnect
```

View TM1 Server configuration

View configuration parameters as defined in the `tm1s.cfg` file for a TM1 server.

Example

Method: GET

Syntax:

```
http://localhost:5657/api/v1/ActiveConfiguration/Administration
```

```
http://localhost:5657/api/v1/StaticConfiguration/Administration
```

Edit TM1 Server configuration

Specify configuration parameters as defined in the `tm1s.cfg` file for a TM1 server.

Example

Method: PATCH

Syntax:

```
http://localhost:5657/api/v1/StaticConfiguration/Administration
Body:
{
  "AdminHost": "xxx.xxx.xxx"
}
```

View TM1 Server loggers

View loggers that are configured for the TM1 server.

Example

Method: GET

Syntax:

```
http://localhost:5657/api/v1/Loggers
```

Edit TM1 Server loggers

Configure a logger for a TM1 server.

Example

Method: PATCH

Syntax:

```
http://localhost:5657/api/v1/Loggers('TM1.Login')
Body:
{
  "Level": "Debug"
}
```

Retrieve number of message log entries

Retrieve the number of entries in the TM1 server message log.

Example

Method: GET

Syntax:

```
http://localhost:5657/api/v1/MessageLogEntries/$count
```

Retrieve message log entries

Retrieve TM1 server message log entries.

Example

Method: GET

Syntax:

```
http://localhost:5657/api/v1/MessageLogEntries?skip=82
```

Adding and deleting users and groups

You can add and delete users and groups on the TM1 Server.

Adding a user

To add a user, complete the following steps.

Procedure

1. Open the Server Explorer.
2. Select the icon for the server you are working with.
3. Right-click the icon and click **Security > Clients/Groups**.

The Clients/Groups dialog box opens.

4. Click **Clients > Add New Client**.

The Creating New Client dialog box opens.

5. Enter the name of the new user in the Enter New Client Name box.

Note: TM1 ignores spaces in user names and passwords. For example, TM1 treats "Mc Cormick" with a space and the name "McCormick" without a space as the same name.

6. Click **OK**.

TM1 adds the user as a new row in the Clients/Groups grid.

When you first add a new user, the user is granted default object security privileges according to the following rules:

- If user groups are already defined for your server, the new user has None privilege to all objects on the server. You must assign the user to a group to allow access to TM1 objects.
- If user groups are not defined for your server, the new user has Write privilege to all objects on the server.

Adding a group

You can add up to 65,535 groups on IBM TM1 Server.

Procedure

1. Open the Server Explorer.
2. Select the icon for the server you are working with.
3. Right-click the server icon, and click **Security > Clients/Groups**.

The Clients/Groups dialog box displays.

4. Click **Groups > Add New Group**.

The Creating New Group dialog box displays.

5. Enter the name of the new group in the Enter New Group Name box.
6. Click **OK**.

TM1 adds the group as a new column in the Clients/Groups dialog box.

Deleting a user

To delete a user, complete the following steps.

Procedure

1. From the Server Explorer, right-click the server icon, and click **Security > Clients/Groups**.
2. In the **Clients/Groups** dialog box, click the user that you want to delete.
3. Click **Clients > Delete Client**.
4. Click **Yes** to confirm the deletion.

Deleting a group

To delete a group, complete the following steps.

Procedure

1. From the Server Explorer, right-click the server icon, and click **Security > Clients/Groups**.
2. In the **Clients/Groups** dialog box, click a cell in the column that represents the group you want to delete.
3. Click **Groups > Delete Group**.
4. Click **Yes** to confirm the deletion.

Assigning users to groups

You can assign users to groups.

Procedure

1. From the Server Explorer, right-click the server icon, and click **Security > Clients/Groups**.
2. In the **Clients/Groups** dialog box, click the check box at the intersection of the user name and the group name.

3. Click **OK**.

Membership in multiple groups

A user who is a member of multiple groups receives the highest level of permissions from all groups.

For example, in the sample data, Ustr3 is a member of two groups.

- North America, which has Write access to the Canada, Mexico, and United States elements in the Region dimension, and Read access to the other elements in the Region dimension.
- South America, which has Write access to the Argentina, Brazil, Chile, and Uruguay elements in the Region dimension, and Read access to the other elements in the Region dimension.

TM1 gives Ustr3 Write access to the Argentina, Brazil, Canada, Chile, Mexico, United States, and Uruguay elements, and Read access to the other elements in the Region dimension.

Securing TM1 data

Because your company uses Planning Analytics for high-level planning and analysis, data can be sensitive and confidential.

Planning Analytics provides all the tools that you need to secure your data, but real security is only as good as the procedures you implement.

Restricting access to the data directory

Using your network file system security, you should always protect the data directory so it is invisible to everyone but the network login used by the server itself.

No matter how elaborate the security on the server, if users can see the data directory, they can view data using TM1 directly from the disk, without connecting to the TM1 Server.

Assigning passwords

Members of the ADMIN group have permissions similar to super users or root users on network systems, they can go anywhere and do anything.

Give out ADMIN passwords carefully, and be sure that ADMIN users know they should not reveal or share their passwords with anyone.

Securing passwords

Treat Planning Analytics passwords the same way as other network passwords.

Be sure that users know not to share passwords. Also, either encourage or require users to change their passwords often.

Setting and clearing passwords

You can set, clear, and change passwords for users.

- Passwords can contain any keyboard character.
- Passwords are not case-sensitive. For example, the password ABC123 is equivalent to abc123.
- Spaces are allowed in passwords, but spaces are ignored by the TM1 Server. The TM1 Server considers the password ABC 123 DEF to be equivalent to ABC123DEF.

Setting a password

You can set up a password.

Procedure

1. From the Server Explorer, right-click the server icon, and click **Security > Clients/Groups**.

2. In the **Clients/Groups** dialog box, click the cell at the intersection of the user name and the Password column.
The cell contains the value of **Undefined**.
3. Type the new password for the user and press **Enter**.
A password can contain a maximum of 256 characters.
TM1 prompts you to retype the new password.
4. Retype the password and click **OK**.
The cell now contains a value of **Defined**.
5. Click **OK**.

Clearing a password

To clear a password, complete the following steps.

Procedure

1. From the Server Explorer, right-click the server icon, and click **Security > Clients/Groups**.
2. In the **Clients/Groups** dialog box, select the cell at the intersection of the Password column and the user name.
3. Click **Clients > Clear Password**.
Confirm whether you really want to clear the password.
4. Click **Yes**.
TM1 clears the password, and displays a value of **Undefined** in the cell.
5. Click **OK**.

Note: When you clear a password, the user can reestablish it the next time they log in to the TM1 Server.

Changing a password

You can change your password any time you are connected to a remote server. Users who are not administrators can do the same.

Procedure

1. In the Server Explorer, select the icon for the server you are working with.
2. Click **Server > Security > Change Password**.
The Password Change dialog box displays.
3. Type the new password in the Password box, and click **OK**.
TM1 prompts you to retype the new password.
4. Retype the password, and click **OK**.

Setting an expiration for a user

When you add a user to TM1, you can set a limit on the number of days the user can access the server.

Procedure

1. From the Server Explorer, right-click the server icon, and click **Security > Clients/Groups**.
2. In the **Clients/Groups** dialog box, in the cell at the intersection of the user name and the Expiration Days column, enter the number of days you want to allow the user to access the TM1 Server.
After the specified number of days, the user will not be able to log in to the server.
3. Click **OK**.

Chapter 6. System and performance monitoring

IBM Planning Analytics includes a collection of tools that provide logging and real-time performance monitoring of the TM1 Servers in your organization. You can monitor the performance and activity of TM1 Server with the following tools.

Planning Analytics and TM1 logs

You can use Planning Analytics logs to monitor the performance of the TM1 Server and the TM1 Admin Server. For example, dimensions and cubes produce messages when they load, and the TM1 Server reports messages when it starts and stops.

You can control the quantity and focus of logging by selectively turning on or off individual TM1 loggers in the logging properties file. The default logging configuration is INFO level messages for all areas of TM1.

Note: You can manage IBM Planning Analytics log files to prevent the accumulation of log files in the TM1 database log directory. Accumulation of log files might cause problems when a TM1 Server or Planning Analytics Administration attempts to access existing log file.

Planning Analytics provides the following log files:

Admin Server log

Log file: tm1admsrv.log in `<pa_install_directory>\bin64\`

Properties file: tm1admsrv-log.properties in `<pa_install_directory>\bin64\`

For more information, see [“The TM1 Admin Server log”](#) on page 42.

Transaction log

Log file: tm1s.log in `<pa_install_directory>\bin64\<database_location>\`

Properties file: tm1s-log.properties in `<pa_install_directory>\bin64\<database_location>\`

For more information, see [“The transactions log”](#) on page 44.

TM1 message log

Log file: tm1server.log in `<pa_install_directory>\bin64\<database_location>\`

Properties file: tm1s-log.properties in `<pa_install_directory>\bin64\<database_location>\`

For more information, see [“The message log”](#) on page 48.

TM1 audit log

Log file: tm1rawstore.<timestamp>

For more information, see [“The audit log”](#) on page 50.

The ops logger Added in Planning Analytics version 2.0.3

Properties file: tm1s-log.properties in `<pa_install_directory>\bin64\<database_location>\`

For more information, see [“The ops logger”](#) on page 55.

TM1 Top log Added in Planning Analytics version 2.0.7

Properties file: tm1s-log.properties in `<pa_install_directory>\bin64\<database_location>\`

For more information, see [“The TM1 Top log”](#) on page 56.

TurboIntegrator process error logs

Log file: TM1ProcessError.log

For more information, see [TM1ProcessError.log file](#).

TM1 event logs

Log file: tm1event.log


For more information, see [Download database log files and EventLogging](#).

Database availability log

Log file: AgentActions.log in `<pa_install_directory>\bin64\<database_location>\`

This log records the user and time associated with these database actions performed in Planning Analytics Administration: start, stop, restart, and end process.

The AgentActions.log file records only actions that are performed through the Planning Analytics Administration user interface. Crashes or actions performed through any means other than the Planning Analytics Administration user interface are not recorded.

Note:  To enable the AgentActions.log on Planning Analytics Local, you must add the following parameter to the bootstrap.properties file for your Planning Analytics Administration agent:

LOG_ACTIONS=TRUE

To learn more about modifying the bootstrap.properties file, see [Install and configure the Planning Analytics Administration agent](#)

The logging properties file

You can use the logging properties file, tm1s-log.properties, to control which log levels are logged for the different components of TM1.

The tm1s-log.properties file is a text file that contains the parameters for configuring logging for a specific TM1 server. For more information, see [“Configuring logging parameters in tm1s-log.properties”](#) on page 39 and [“Configuring logging level in tm1s-log.properties”](#) on page 41.

Each TM1 server uses its own tm1s-log.properties file and checks for the file whenever the server is started. After startup, the server checks for updates to the properties file and adjusts the logging as needed. You can update a properties file in real time and the TM1 server dynamically reads the changes and adjusts the logging.

Note: The TM1 local server checks only the tm1s-log.properties file when the local server starts up.

A sample tm1s-log.properties file is provided in the directory of some sample TM1 databases that you install.

Example tm1s-log.properties file

```
log4j.logger.TM1=INFO, S1

# S1 is set to be a SharedMemoryAppender
log4j.appender.S1=org.apache.log4j.SharedMemoryAppender
# Specify the size of the shared memory segment
log4j.appender.S1.MemorySize=5 MB
# Specify the max filesize
log4j.appender.S1.MaxFileSize=100 MB
# Specify the max backup index
log4j.appender.S1.MaxBackupIndex=20
# Specify GMT or Local timezone
log4j.appender.S1.TimeZone=GMT

log4j.logger.Event=INFO, S_Event
log4j.appender.S_Event=org.apache.log4j.SharedMemoryAppender
log4j.appender.S_Event.MemorySize=1 MB
log4j.appender.S_Event.MaxFileSize=100 MB
log4j.appender.S_Event.MaxBackupIndex=1
log4j.appender.S_Event.File=tm1event.log
log4j.appender.S_Event.Format=TM1Event
log4j.appender.S_Event.TimeZone=GMT
```


Enabling and disabling logging

Logging is enabled when a TM1 server detects a logging properties file in the same directory where the server's configuration file, `tm1s.cfg`, is located.

Procedure

1. Edit a copy of the sample logging properties file, `tm1s-log.properties`, to configure logging message level and output settings, as described in “[Configuring logging level in tm1s-log.properties](#)” on page 41 and “[Configuring logging parameters in tm1s-log.properties](#)” on page 39.
2. Put the `tm1s-log.properties` into the same directory where the configuration file, `tm1s.cfg`, is located for the TM1 database that you want to monitor.

The location of the `tm1s.cfg` file is typically the data directory of the TM1 database, but could be a different location depending on your specific TM1 configuration. For example, if your `tm1s.cfg` file is located in the `C:\salesdata` directory on a Microsoft Windows system, then copy the `tm1s-log.properties` file into this directory.

After the `tm1s-log.properties` file is placed in the same directory as the `tm1s.cfg` file, the server starts logging based on the settings configured in the `tm1s-log.properties` file. It is not necessary to restart the TM1 server to initialize logging, unless you are running a local server.

Note: The TM1 local server only checks the logging properties file, `tm1s-log.properties`, when the local server starts up. After startup, a local server never checks for changes in the `tm1s-log.properties` file, so all settings in a local server's `tm1s-log.properties` file must be considered static. If you change any logging properties for a local server, you must restart the server to apply the new logging settings.

3. To temporarily turn off logging, set the TM1 logger values to either INFO or OFF. Using a value of INFO is recommended.
 - Setting a logger value to INFO will continue some logging, but at a much decreased amount compared to a setting of DEBUG. Using a setting of INFO, instead of OFF, is useful because TM1 will still log important messages for WARN, ERROR, and FATAL messages.
 - Setting a logger value to OFF will stop all logging for that logger, and you could miss any potential WARN, ERROR and FATAL messages.

Configuring logging parameters in `tm1s-log.properties`

Logging output is configured in the appender section of the `tm1s-log.properties` file with the following parameters.

Parameter	Description
MemorySize	<p>Specifies the size of the shared memory segment. This memory represents the area of system memory where log messages are sent before being written to the log file.</p> <p>Default value is 5 MB.</p> <p>Use the following format:</p> <pre>log4j.appender.S1.MemorySize=Size Units</pre> <p>where:</p> <ul style="list-style-type: none">• Size is the numeric value for the memory size.• Units can be KB, or MB. <p>For example, to set the MemorySize to 5 MB, enter the following:</p> <pre>log4j.appender.S1.MemorySize=5 MB</pre>

Parameter	Description
MaxFileSize	<p>Specifies the maximum file size that the log file is allowed to take up on disk.</p> <p>Default size is 100 MB.</p> <p>Uses the following format:</p> <pre>log4j.appender.S1.MaxFileSize=SizeUnits</pre> <p>where:</p> <ul style="list-style-type: none"> • Size is the numeric value for the file size. • Units can be KB, MB, or GB. <p>For example, to limit the log file size to 10 MB, enter the following:</p> <pre>log4j.appender.S1.MaxFileSize=10 MB</pre> <p>If the MaxBackupIndex parameter is set to 1 or greater, then the logging process automatically creates a backup file when the log file reaches the MaxFileSize. The total number of backup files is determined by MaxBackupIndex option.</p>
MaxBackupIndex	<p>A numeric value that specifies how many backup files are kept before the oldest log file is erased.</p> <p>Default value is 20.</p> <ul style="list-style-type: none"> • If set to 0 - no backup log files will be saved and the current log file will be overwritten and restarted each time it reaches MaxFileSize. • If set to 1 or higher - A total of x backup files will be maintained, where x is equal to the MaxBackupIndex value. <p>The backup process repeats each time the MaxFileSize is reached, overwriting the oldest backup file so that only the newest backup files remains.</p>
TimeZone	<p>Specifies the timezone for the message timestamps in the log file. Uses the following format:</p> <pre>log4j.appender.S1.TimeZone=Zone</pre> <p>where Zone can be set to either GMT or Local.</p> <ul style="list-style-type: none"> • If set to Local, the log messages will have timestamps based on the local time of the machine where the TM1 server is running. • If set to GMT, the log messages will have timestamps based on Greenwich Mean Time. • If the parameter is set to something other than GMT or Local, or if the parameter is not set at all, it defaults to GMT.

Parameter	Description
File	<p>Specifies the log file.</p> <p>For example:</p> <pre>log4j.appender.S2.File=tm1event.log</pre> <p>where the filename is tm1event or tm1top.log</p> <p>For example, to generate critical log messages using the ops logger:</p> <pre>log4j.logger.Event=INFO, S2 log4j.appender.S2=org.apache.log4j.SharedMemoryAppender log4j.appender.S2.MemorySize=5 MB log4j.appender.S2.MaxFileSize=100 MB log4j.appender.S2.MaxBackupIndex=1 log4j.appender.S2.TimeZone=Local log4j.appender.S2.File=tm1event.log log4j.appender.S2.Format=TM1Event</pre>
Format	<p>Specifies the format of the logs.</p> <p>For example:</p> <pre>log4j.appender.S2.Format=TM1Event</pre> <p>where the format is TM1Event or TM1Top</p> <p>For example, to dynamically monitor the threads that are running and outputs thread status to the tm1top.log:</p> <pre>log4j.logger.Top=INFO, S_Top log4j.appender.S_Top=org.apache.log4j.SharedMemoryAppender log4j.appender.S_Top.MemorySize=5 MB log4j.appender.S_Top.MaxFileSize=100 MB log4j.appender.S_Top.MaxBackupIndex=20 log4j.appender.S_Top.File=tm1top.log log4j.appender.S_Top.Format=TM1Top</pre>

Configuring logging level in tm1s-log.properties

Use the following two statements in the logging properties file as the standard configuration to log all INFO level logs for all areas of TM1.

```
log4j.rootLogger=INFO, S1
```

```
log4j.logger.TM1=INFO
```

Use the following format to configure logging at a specific message level:

```
TM1 logger=Message level, Appender
```

Where:

- *TM1 logger* is the name of the TM1 subcomponent that you want to log. Contact customer support for assistance using TM1 loggers.
- *Message level* is the message severity level that you want to log. Valid values include: DEBUG, INFO, WARN, ERROR, FATAL, or OFF, as described in the section "Message Severity Levels."
- *Appender* is the output destination. Use a value of S1 for the TM1 shared memory appender which transfers messages to memory before saving them to a file.

Message levels are logged as follows:

- Setting logging to DEBUG will report all severity levels messages.
- Setting logging to WARN will report WARN, ERROR, and FATAL messages.
- Setting logging to OFF disables all logging for the specific TM1 logger.

For example, to turn on logging at the DEBUG level for all TM1 sub-components, you would use the following statements:

```
log4j.rootLogger=DEBUG, S1
```

```
log4j.logger.TM1=DEBUG
```

The TM1 Admin Server log

The Admin Server log file contains messages about communication between clients, the Admin Server, and individual TM1 Servers. The TM1 Admin Server log is useful for troubleshooting connection issues when you use the TM1 Transport Layer Security (TLS) with custom certificates or certificates from the Microsoft Windows certificate store.

Most of the messages in the Admin Server log are created during the startup process for clients and servers. Messages are logged when:

- TM1 servers start up and register with the TM1 Admin Server.
- TM1 clients contact the TM1 Admin Server for a list of available TM1 servers.
- SSL security is established between TM1 clients, the TM1 Admin Server, and individual TM1 servers.

See [Planning Analytics Local security](#) in the *Planning Analytics Local Installation and Configuration* documentation.

Enabling Admin Server logging

Logging for the TM1 Admin Server is enabled by default when you install TM1.

The installation places the `tm1admsrv-log.properties` file and the TM1 Admin Server program file, `tm1admsrv.exe`, into the `<pa_install_directory>\bin64` directory.

Logging is activated when the TM1 Admin Server starts up and detects the `tm1admsrv-log.properties` file in the same directory.

Configuring Admin Server logging

Configuration for TM1 Admin Server logging is contained in the `tm1admsrv-log.properties` file. This file is located in the `<pa_install_directory>\bin64` directory.

The sections of the `tm1admsrv-log.properties` file are:

• **Logger section**

Configures the TM1 subcomponents and message severity level that you want to log.

Example:

```
log4j.rootLogger=INFO*, R1
```

```
log4j.logger.TM1=INFO
```

*INFO is the message level.

• **Appender section**

Example:

Controls the output destination of the logging. The default configuration uses the `RollingFileAppender` option to write messages to the `tm1admsrv.log` file.

```
# R1 is set to be a RollingFileAppender
```

```
log4j.appender.R1=org.apache.log4j.RollingFileAppender
```

```
log4j.appender.R1.File=tm1admsrv.log*
```

```
log4j.appender.R1.MaxFileSize=10 MB
```

```
log4j.appender.R1.MaxBackupIndex=2
```

*tm1admsrv.log is the log file name.

• Pattern Layout section

Controls the output fields and formatting of the messages that are written to the log file. The default settings use a time reference of GMT.

Example:

```
# R1 uses PatternLayout
```

```
log4j.appender.R1.layout=org.apache.log4j.PatternLayout
```

```
log4j.appender.R1.layout.ConversionPattern=%t %p %d(%Y-%m-%d %H:%M:%S,%Q) %c  
%m%n
```

```
log4j.appender.R1.layout.TimeZone=GMT
```

The default logging configuration logs all INFO level messages. You can adjust the logging message level by editing the following two statements in the tm1admsrv-log.properties file:

```
log4j.rootLogger=INFO, R1  
log4j.logger.TM1=INFO
```

For example, replace INFO with the DEBUG logging level:

```
log4j.rootLogger=DEBUG, R1  
log4j.logger.TM1=DEBUG
```

Note: The default logging configuration is intended for everyday use and does not typically require adjustment. Contact customer support for assistance if you need to configure the logging properties file for troubleshooting purposes.

Message severity levels for Admin Server logging

The Admin Server log categorizes messages into three severity levels. These levels are also used in the tm1admsrv-log.properties file to configure logging to a specific level.

Parameter	Description
DEBUG	Detailed, technical messages that are useful when customer support or development engineering need to debug the application. When logging is configured to this level, DEBUG, INFO, and ERROR messages are logged.
INFO	Informational messages that highlight the progress of the application and report normal transitions within the application. When logging is configured to this level, INFO and ERROR messages are logged.
ERROR	An error condition of which you should be aware. Action should be taken to fix or report the issue to customer support. When logging is configured to this level, only ERROR messages are logged.

Viewing the Admin Server log file

The default logging configuration writes log messages to the tm1admsrv.log file. The log file is an ASCII text file that you can open in any text editor, such as Microsoft Windows Notepad.

Procedure

1. Locate the tm1admsrv.log file in the <pa_install_directory>\bin64 directory.
2. Open and view the log file with a text editor, such as Microsoft Windows Notepad.

Each line in the log file represents one unique message, arranged in the following format:

Thread ID	Message Level	Date and Time	Subcomponent Name	Message Text
3948	INFO	2007-01-19 19:10:22, 128	TM1.Comm.SSL	Error connecting to remote machine
1260	DEBUG	2007-01-19 19:14:04, 396	TM1.Server	Logger initialized
3328	DEBUG	2007-01-19 19:14:04, 396	TM1.Event	Timwed wait for event 0043B858 succeeded
3328	DEBUG	2007-01-19 19:14:04, 396	TM1.Event	Destroy event 0043B858 succeeded
1260	DEBUG	2007-01-19 19:14:04, 396	TM1.Event	Signal event 0043B858 succeeded

The transactions log

Each TM1 database logs the data changes that are made by its clients, such as changes to cube data, in the tm1s.log file. This logging cannot be disabled at the database level. Cube logging can be enabled and disabled for specific cubes. Transaction log files can be used for auditing and data recovery.

When a client changes a cube value, TM1 records the change in the tm1s.log file, which is located in the TM1 server data directory, for example, `<pa_install_directory>\bin64\samples\tm1\SData\tm1s.log`.

The following information is recorded in the tm1s.log file:

- Date and time the change was made
- Name of the client who made the change
- Whether the new data is simple data (N) or string data (S)
- Value before the change
- Value after the change
- Name of the cube in which the change was made
- Elements that identify the cell that changed

Consider the following scenarios that impact transaction logging:

- Transaction logging can be enabled and disabled for specific cubes by setting the LOGGING column in the }CubeProperties cube. Transaction logging is typically needed on cube transactions only during period of manual user data entry. For more information, see [“Enabling and disabling transaction logging”](#) on page 45.
- Transaction logging can be enabled and disabled during TurboIntegrator processes by using the CubeSetLogChanges() TurboIntegrator function. You might want to disable transaction logging during a TurboIntegrator process that loads a large volume of data to a cube. For more information, see [CubeSetLogChanges](#).
- Consider whether transaction logging is required during TurboIntegrator processes. In general, if a TurboIntegrator process can be rerun to reload the same data into a cube, transaction logging can be disabled during the TurboIntegrator process.
- The current transaction log file is named tm1s.log. The tm1s.log file is locked by the TM1 database and cannot be viewed or edited with a text editor. The current transaction log file is renamed to tm1s<timestamp>.log during a SaveDataAll (SaveDataAll happens automatically on shutdown or

restart). An empty tm1s.log file is generated when the TM1 database starts and during a SaveDataAll without a TM1 database restart. For more information, see [SaveDataAll](#).

- The transaction log files are not managed by the TM1 database because these log files contain a record of changes to data that might be required for review. It's possible for the transaction log files to grow large.
- Consider how long to keep the historical transaction log files (historical files are named tm1syyyymmddhhmmss.rej). This length of time can be determined based on the requirement to store changes. If the historical transaction logs files must be permanently stored, then it is recommended to either compress transaction log files older than a specific date, or to move the files to a storage solution. The historical transaction logs files can be opened and examined in a text editor. A text editor that can open large files is recommended.
- If the server is intentionally shut down without saving the changes, the log file is saved with a timestamp and the extension is changed to .rej. You can process the tm1s<timestamp>.log file through TurboIntegrator to recover the transactions.
- During an improper server shutdown, the tm1s.log remains as is and is not timestamped during shutdown. When the server restarts, it recognizes the presence of an old tm1s.log file and prompts you to choose between recovering the new log from the old existing log. If you respond to recover, the server searches for the end of the same file (tm1s.log) and resumes logging to it; but if you reject, the server creates a tm1s<timestamp>.rej file and starts logging to a new tm1s.log.

The log file is a comma-delimited ASCII file, as shown in the following sample:

```
"19980602212741", "19980602212741", "Admin", "N", "380.", "250.", "salescube",  
"Budget", "Belgium", "L Series 1.6 L Sedan", "Units", "Jun", ""  
"19980602212744", "19980602212744", "Admin", "N", "430.", "600.", "salescube",  
"Budget", "Belgium", "L Series 1.6 L Sedan", "Units", "Sep", ""  
"19980602212749", "19980602212749", "Admin", "N", "610.", "800.", "salescube",  
"Budget", "Belgium", "L Series 1.6 L Sedan", "Units", "Oct", ""
```

For more information, see also [TM1 Server overview](#).

Enabling and disabling transaction logging

By default, TM1 logs transactions to all cubes loaded on the server. As the system administrator, you have the option to turn off logging for particular cubes. When you disable logging, TM1 accelerates updating the data but you cannot recover the updates in the event of a system failure.

Procedure

1. Open Server Explorer.
2. Select the Cubes icon for the server you are working with.
3. Select **Cubes > Security Assignments**.

The TM1 Security Assignments dialog box opens.

4. Click the cell at the intersection of the Logging column and the cube name.

TM1 enables logging when a check box contains an **X**, and disables logging when the check box is empty.

5. Click **OK**.

Viewing the transaction log

You can query the transaction log (tm1s.log) to view the records of all the logs currently in the TM1 server data directory. When you query the transaction log, TM1 combines all the log files into one logical file that satisfies the query parameters. For example, if you query for all the records starting on Jan. 2, 2002 at 2:30 PM GMT, TM1 returns all the records in all the transaction logs with a time stamp of 20020102143000 or later.

Procedure

1. Select the server in Server Explorer.
2. Click **ServerView Transaction Log**.

The Transaction Log Query dialog box opens.

3. Click the right arrow in a parameter field to set the parameters for the query.

There are four parameters you can set:

Parameter	Description
Starting Time	Queries all the records written to the transaction log on or after the starting time. Start date and time for the query. The date and time format is MM/DD/YYYY HH:MM:SS GMT. The default start date and time is 00:01:00 GMT on the date you launch the query.
End Time	End date and time for the query. The date and time format is MM/DD/YYYY HH:MM:SS The default is __/__/______:__:__, which is an open end date and time. If you accept the default, TM1 queries all the records up to the time you launch the query.
Clients	The client(s) against which you apply the query. You can query against either a single client or all the clients. The default is all clients (*).
Cubes	The cube(s) against which you apply the query. You can query against either a single cube or all the cubes. The default is all cubes (*)

4. Click **OK**.

The query returns a table with all the transaction records that satisfy the parameters you set. The table displays in the Transaction Log Query Results dialog box.

By default, records are sorted in ascending order by LOGTIME.

5. To sort on a different column, click the column heading. To change the order of a column sort, click the column heading a second time.
6. Click **Edit > Find** to search the records in the query results table.

Setting search lock limit

When the transaction log is searching, other users are locked out of performing activity on the log. By default the log searches 5000 lines before it temporarily releases the lock so that other users can access the log.

You can change the number of lines searched before the lock is temporarily released using the LogReleaseLineCount parameter set in the tm1s.cfg file. For more information, see [Parameters in the tm1s.cfg file](#) in the *Planning Analytics Installation and Configuration* documentation.

Data backup and recovery

When you bring down the server normally, TM1 renames the log file by appending a time stamp to it, using the following naming convention: tm1s yyyyymmddhhmmss.log. The tm1s.log file remains open while a TM1 server is running.

The time stamp, *yyyyymmddhhmmss*, represents the current Greenwich Mean Time at the time the server was brought down. For example, if the server came down on January 2, 2002, at 2:30 PM, the name of the log file is tm1s20020102143000.log.

TM1 recovers the data automatically in the event that a server comes down abnormally, and leaves the tm1s.log file on the disk. The next time you bring up the server, TM1 recovers the changes in either of two ways:

- **Automatically recovers the changes** -- when you are running the TM1 server as a Microsoft Windows service.
- **Prompts you to recover the changes** -- when you are running the TM1 server as an application

If you intentionally shut down a TM1 server without saving the data, TM1 saves the transaction log with a time stamp and changes the file extension to .rej. For example, tm1s20020102143000.rej. The .rej log file ensures that you always have a record of the data transactions, even if you shut down the server without saving the data. If you accidentally shut down the server without saving the changes, you can process the .rej file through TurboIntegrator to recover the data.

Backing out transactions using the transaction log

After you query the transaction log, you can use the Transaction Log Query Results dialog box to back out the transactions. When you back out a transaction, the value in the OLDVALUE column replaces the value in the NEWVALUE column.

Procedure

1. Highlight one or more transactions that you want to back out.
 - To highlight an individual transaction, click the transaction.
 - To highlight multiple adjacent transactions, click the first transaction and SHIFT+click the last transaction.
 - To highlight multiple nonadjacent transactions, CTRL+click each transaction.

2. Click **Edit, Select**.

All the highlighted transactions now display a check mark in the box beside the first column. The check marks indicate that the transaction is selected to be backed out.

To select all the transactions without first highlighting them, click **Edit, Select All**.

3. Click **Edit, Back Out**.

TM1 backs out the transactions in reverse chronological order as identified by the LOGTIME column.

Removing transaction log files from the disk

The TM1 log files can take up a substantial amount of disk space after the server has been running for some time.

You should remove the old log files from your disk every so often, depending on the volume of the changes you make and the size of your disk. You can back up these files before you erase them.

Do not remove the log files when the TM1 server is running. First shut down the server, and then delete the log files from your disk.

Troubleshooting: Recovering from a corrupted transaction log file

In some cases, an unexpected or incomplete shutdown of the TM1 server, due to a server crash or power outage, can cause the transaction log file to become corrupted. If this happens, the server will not be able to restart.

In this case, you need to troubleshoot the cause of the shutdown, verify that the transaction log file is corrupted, remove the log file and contact Customer Support for assistance.

Determining whether the transaction log file is corrupted

When the TM1 server restarts after an unexpected shutdown, it attempts to recover data from the transaction log file, tm1s.log. If TM1 detects that the log file is corrupted, the server will not start and informs you of the situation with visual prompts or logged messages. The exact type of message depends on how you are running the TM1 server:

- On a UNIX system or as a Microsoft Windows service.
- As a Microsoft Windows application.

Resolving a corrupted transaction log file

To resolve this condition, move the transaction log file, tm1s.log, from the TM1 server data directory to a temporary location on your system. Now, you can successfully restart the server, but you can also contact Customer Support for assistance in resolving the cause of the unexpected shutdown.

TM1 Server running on a UNIX System or as a Microsoft Windows service

If you are running the TM1 server on a UNIX system or as a Microsoft Windows service, TM1 cancels the automatic recovery when it tries to restart the server and writes a warning message to the TM1 server message log, tm1server.log. This message indicates that the transaction log is corrupted. An example messages in the TM1 Server Message log indicating a corrupted transaction file and the warning messages are shown in the following table.

Thread ID	Message Level	Date and Time	Subcomponent Name	Message Text
4076	FATAL	2007-10-30 18:20:10, 203	TM1.Server	Bad transaction log record: 828.7...
4076	FATAL	2007-10-30 18:20:14, 921	TM1.Server	Server terminated

TM1 Server running as a Microsoft Windows application

If you are running the TM1 server as a Windows application, TM1 first prompts you to recover the changes when the server restarts.

If you select **No**, the server continues running, but the problem that caused the unexpected shutdown might still exist.

If you select **Yes**, TM1 displays a warning message and also writes a message to the TM1 server message log.

Select **OK** to shut down the TM1 server.

The message log

The TM1 Server records status messages about the activity of the server in the tm1server.log file.

These messages contain details on activity such as executed processes, chores, loaded cubes and dimensions, and synchronized replication.

The TM1 Server message logging system is designed to minimize the impact on performance while allowing greater control over the quantity and focus of data produced by logging.

The logging system includes the following components:

- Message severity levels - Categories for classifying and reporting messages by severity; DEBUG, INFO, WARN, ERROR, and FATAL.
- TM1 loggers - Parameters that provide enhanced control for selectively logging specific areas or sub-components of TM1.
- Logging properties file - A text-based file that enables you to configure and enable logging for a specific TM1 server.
- Message log viewer - A tool for viewing the message log in Server Explorer and Architect.

Note: Log messages are displayed in English only.

The message log is automatically managed by the TM1 database. By default, the maximum log file size is 100 MB. When the log file grows larger than 100 MB, it is rolled over automatically by the TM1 database. The tm1server.log file is renamed tm1server.log.1, and an empty tm1server.log file is created.

By default, a maximum of 20 TM1 database message log files are maintained by the TM1 database (tm1server.log, tm1server.log.1,...tm1server.log.19). Maximum log file size and maximum number of historical files are based on configuration settings in the tm1s-log.properties file.

Configuring message logging

Use a copy of the logging properties file, tm1s-log.properties, to configure and enable message logging for a specific TM1 server.

For more information, see [“The logging properties file”](#) on page 38.

Message severity levels

The logging system categorizes messages into severity levels.

Parameter	Description
DEBUG	Detailed, technical messages that are useful when customer support or engineering need to debug the application. Examples: "SSL Connection opened", "Removing SQL Connection from memory"
INFO	Informational messages that highlight the progress of the application and report normal transitions within the application. Examples: "Server is ready", "Loading dimension..."
WARN	A potentially harmful situation or condition of which you should be aware. Action may be required, but operation of the server is not interrupted. Example: "Attempt to add a new group exceeded the limit on group creation"
ERROR	An error condition of which you should be aware. Action should be taken to fix or report the issue to customer support. The error may be so severe that the server shuts down or it may result in the current client request being canceled. Examples: "Error connecting to remote machine", "Failed to create a cube"
FATAL	A very severe error event that will possibly cause the server to shut down or result in the current client request being canceled. You should immediately take action to fix the issue and report the event to customer support.

Viewing the TM1 Server message log

To view a list of the messages that the TM1 server has recorded, complete the following steps.

Procedure

1. Select the server in the Server Explorer.
2. Click **Server, View Message Log**.

The Message Log window opens.

The message log fields are defined in the following table.

Message Field	Description
Thread ID	ID number of the thread in the TM1 server that generated the logging event. Example: 2488

Message Field	Description
Message Level	Severity level of the message being reported: DEBUG, INFO, WARN, ERROR, or FATAL.
Date and Time	Date and time the message was logged, based on the TimeZone parameter setting in the logging properties file. Data and time can be recorded as either GMT or Local. For more details, see “Configuring logging parameters in tm1s-log.properties” on page 39. Reported in the format: yyyy-mm-dd hh:mm.ss,milliseconds Example: 2006-10-02 16:49.01,925
TM1 Logger Name	Name of the TM1 sub-component that generated the message or activity. Example: Start time: Tue Dec 05 2006 11:02:47
Message Text	Pre-defined text message that describes the error or activity being reported.

- If a message in the log shows an error condition for an executed process or replication, you can double-click the message to view the details of why the activity generated the error.

Searching for text in the message log

You can search for text in the message log.

Procedure

- Click **Edit > Find** or press CTRL+F.

The Find dialog box opens.

- Enter the text you to find and then click **Find Next**.

If the text is found, the line containing the matched text is highlighted and shown in the Message Log window.

Copying text from the message log

You can copy text from the message log.

Procedure

- Select the line of text you want to copy.
- Select **Edit > Copy** or press CTRL+C to copy.

The currently selected line is copied to the Microsoft Windows clipboard and can then be pasted into other applications.

The audit log

The TM1 audit log monitors changes to metadata, such as modifications to dimensions, views and subsets. It also tracks login and TurboIntegrator process execution.

This log can be used for accountability purposes where laws or regulations, such as Sarbanes-Oxley, require auditing of certain activities in mission critical software.

Each TM1 server maintains its own separate audit log. Administrators can use a server's audit log to answer these questions:

- What object was changed?

- How was the object modified?
- Who made the change?
- When was the change made?

By default, audit logging is disabled and must be manually enabled for each server you want to monitor. Audit logging is only on or off. It is not possible to enable audit logging for only some activity types. For more information, see [“Configuring audit logging”](#) on page 52.

The active audit log file is locked and cannot be opened in a text editor. Audit logging is enabled and disabled with the AuditLogOn parameter in tm1s.cfg file.

You can query and view the audit log using the Audit Log window, available in TM1 Server Explorer. The log file is rolled over by the TM1 database on restart. Historical audit log files are named tm1auditstore<timestamp>.log.

Audit logging is not managed automatically by the TM1 database. If audit logging is enabled, you should determine how long audit logs are kept in the logging directory. The audit log files might be large.

Understanding audit log events

The audit log contains records of changes to major TM1 objects and system-wide activity. These records are called events.

TM1 object events

The audit log monitors activity for the TM1 objects. A common set of events, such as object created or object deleted, are logged for all objects. Additionally, events that are specific to each object are logged. For example, the execution of a process is an event that is logged only for process objects.

The following objects are monitored:

- Applications
- Chores
- Clients
- Cubes
- Dimensions
- Elements
- Groups
- Processes
- Rules
- Subsets
- Views

System-wide events

System-wide events include operations related directly to the TM1 server such as successful and failed login attempts, server startup and shutdown, and changes to server parameters.

Audit log messages for dimension sorting events

When you sort or change the order of elements in a dimension, the exact message recorded in the audit log depends on the sort method that was used; automatic or manual.

Automatic Dimension Sorting

When you set an automatic type sort for a dimension, the audit log records a specific descriptive message about the action. An automatic sort is performed using the **Dimension Element Ordering** dialog box available when you right-click on a dimension and select **Set Elements Order**.

For example, the following message is reported in the audit log for an automatic sort change to the account dimension:

```
"136","account","DESCENDING","DimensionSortElementsSense  
property set for Dimension 'account': DESCENDING"
```

The Audit Log window displays the following message:

```
Property SortElementsSense was set to 'DESCENDING' for dimension 'account'.
```

Manual Dimension Sorting

When you edit the order of a dimension in manual mode using the Dimension Editor, the audit log records only a generic message. Manual mode includes changing the element order by hand, or selecting Sort from the Edit menu in the Dimension Editor, and clicking the Set Dimension order button followed by saving the dimension.

For example, the following generic message is reported in the audit log for a manual sort change to the account dimension:

```
21,account,Dimension updated: account
```

The Audit Log window displays the following message:

```
Dimension 'account' was updated.
```

Configuring audit logging

By default, audit logging is disabled. You must manually configure audit logging using the audit log parameters in the TM1 server's configuration file (tm1s.cfg).

To enable audit logging for a specific TM1 server, complete the following steps.

Procedure

1. Open the tm1s.cfg file for that server.
2. Set the following parameters in the tm1s.cfg file:

```
AuditLogOn=T
```

```
AuditLogUpdateInterval=60
```

3. Restart the server.

Note: After changing the value for the AuditLogOn parameter, you must restart the TM1 server for the new value to be applied.

For more details about the tm1s.cfg file and the audit log parameters, see [Parameters in the tm1s.cfg file in the *Planning Analytics Installation and Configuration* documentation.](#)

Updating the audit log with the latest events

When audit logging is enabled, TM1 runs the logging in the background and automatically updates the audit log at a set interval. This interval is determined by the AuditLogUpdateInterval parameter in the tm1s.cfg file.

You can manually update the audit log whenever you want by using the Process Audit Log Events command in TM1 Server Explorer. This action updates the audit log to include the latest events and is especially useful to do just before you open the Audit Log window to run queries.

Procedure

1. Select a TM1 server in Server Explorer.
2. Click **Server > Process Audit Log Events**.
3. Click **Yes** to confirm the update process.

The audit log now contains the latest audit event records.

Using the audit log window to view log messages

The content of the audit log is accessible through the Audit Log window, available in TM1 Server Explorer.

You use the Audit Log window to query the audit log and to view the audit log event messages retrieved by the query.

The Audit Log window contains two main panels:

- **Query panel** - Use the Query panel to build queries that search the audit log for a specific time period or type of event.
- **Results panel** - Use the Results panel to view and navigate the records retrieved by your query.


Opening the audit log window

You can open the Audit Log window.

Procedure

1. Select a TM1 server in Server Explorer.
2. Click **Server, View Audit Log**.

The Audit Log window opens.

You can then select the query parameters that you want and click the **Run Query** button  to retrieve the records for the query.

Creating and running queries

Use search parameters in the Query panel of the Audit Log window to narrow your search.

Procedure

1. Set the **Date and Time** option.


This option includes pre-defined time periods including today, and the last 10, 30, 60 and 90 days.

If you are looking for events from a specific time period, select **Custom Time Period** from the **Time Period** drop-down box. Enter a start and end time.

2. Set the **Event Owner** option.

This option asks the question "Who caused this event?" This can be an actual TM1 user or a scheduled chore.

- Click **All** to search for events caused by any user or any scheduled chore.
- Click **Client** to search for events caused only by a user.
- Click **Scheduled Chore** to search for events caused only by a scheduled chore.

To select a specific user or scheduled chore, click the Select button  next to the related field.

The **Select Client** or **Select Chore** dialog box opens.

- To select an individual user or chore name, click the item.
- To select an adjacent range of user or chore names, click the first item in the range, hold down SHIFT, and click the last item in the range.
- To select multiple nonadjacent user or chore names, hold down CTRL and click each item.

3. Set the **Event Type** option.

Use this option to select the exact type of event for which you want to search. For example, "find unsuccessful login attempts" or "find events where a dimension was deleted".

- Click **All** to search for both types of audit events; system-wide and object related events.
- Click **System-wide** to search for only system-wide audit events.

To search for a specific system-wide event, select the event from the **Event** list.


The default setting, *, searches for all system-wide events.

- Click **Object** to search for events related to only TM1 objects.


The **Object** option has the following sub-options:

- Set the **Object Type** option to limit the query to only a specific type of TM1 object. For example, events related only to dimensions.
- Set the **Object Name** option to find events for a specific object name.
- Set the **Event** option to search for a specific object event.

The list of events changes depending on which type of object you select in the **Object Type** drop-down box.

Click the Select Object Name button  next to the **Object Name** field to display a dialog box where you can select objects by name.

- To select an individual object name, click the item.
- To select an adjacent range of object names, click the first item in the range, hold down SHIFT, and click the last item in the range.
- To select multiple non-adjacent object names, hold down CTRL and click each item.

Note: When you set Object Type to **Element**, the Select Object Name button  becomes disabled because the element list could be too large to display. If you want to search for events related to a specific element, you must manually enter an element name using the following format:

```
DimensionName:ElementName
```


For example: region:italy

4. Click the **Run Query** button .

The records retrieved by the query appear in the **Results** panel grid.

Viewing results


The Results panel grid organizes the audit log records from the query into the following columns.

Message Field	Description
Date	Date and time of the event.
User	TM1 client (user) or scheduled chore that was responsible for causing the event.
Event Type/Description	Brief description of the event.
Object Type	Type of TM1 object associated with the event.
Object Name	Name of the TM1 object associated with the event.
Details	Displays an icon to indicate that detailed information exists for the specific event. If an event has details, you can view the details by clicking on the Details  for that record.

You can sort the results in the grid in ascending or descending order for any column by clicking on the column title.

Viewing event details

Some events have additional information stored as event details.

If an event has details, a Details icon  is displayed in the Details column for that event.


To view the details for an event, click the Detail icon for that event.

Event details display in the Audit Log Details window, separate from the main Audit Log window. You can open multiple Audit Log Details windows at a time to compare them side-by-side.

Copying data to the Windows clipboard

You can copy event data from the grid to the Windows clipboard. The copy feature is available in both the Audit Log and Audit Log Details windows.task_it


Procedure

To copy event data, highlight an individual cell in the Results grid and then click the **Copy** button  on the Results toolbar.

Using the find feature

The Results panel includes a Find tool that searches for specified text in the Results grid.

Procedure

1. Click **Find**  in the Results toolbar.

The Find dialog box appears.

2. Enter the text you want to find
3. Click **Find Next**.

If the text is found, the cell containing the matching text is highlighted in the Results grid.

Exporting results

The Results panel also includes an export option for exporting results in XML, comma delimited, and tab delimited file formats. The export feature is available in both the Audit Log and Audit Log Details windows.

Procedure

1. Click **Export**  in the Results toolbar.

The Save As dialog box appears.

2. Enter a filename and location for the file.
3. Select the file format using the **Save as type** option.
 - XML (*.xml)
 - CSV (Comma delimited) (*.csv)
 - Text (Tab delimited) (*.txt)
4. Click **Save**.

The ops logger

Added v2.0.3 The TM1 server uses monitoring code to generate log messages of a critical nature that are pushed out to the ops.logger.

Critical events that are pushed out to this logger include the following:

- Threads and their state
- Thread state of interest
- High contention

- Pool memory size
- Chore failure

Event logging is enabled by using the `EventLogging` configuration parameter in the `tm1s.cfg` file. When event logging is enabled but there is no event logger defined, the TM1 server modifies the `tm1s-logs.properties` file to add the definitions of the event logger.

The following properties are added to the `tm1s-log.properties` file:

```
log4j.logger.Event=INFO, S2
log4j.appender.S2=org.apache.log4j.SharedMemoryAppender
log4j.appender.S2.MemorySize=5 MB
log4j.appender.S2.MaxFileSize=100 MB
log4j.appender.S2.MaxBackupIndex=1
log4j.appender.S2.TimeZone=Local
log4j.appender.S2.File=tm1event.log
log4j.appender.S2.Format=TM1Event
```

And the following settings are added to the `tm1s.cfg` file:

```
EventLogging=T
EventScanFrequency=1
EventThreshold.PooledMemoryInMB=0
EventThreshold.ThreadBlockingNumber=5
EventThreshold.ThreadRunningTime=600
EventThreshold.ThreadWaitingTime=20
```

These settings are defined as follows:

EventScanFrequency

Specifies the period to check the collection of threads, where 1 is the minimum number and 0 disables the scan.

EventThreshold.PooledMemoryInMB

Specifies the threshold for which a message is printed for the event that the server's pooled memory exceeds a certain value.

EventThreshold.ThreadBlockingNumber=5

Indicates that a warning is printed when a thread blocks at least five other threads.

EventThreshold.ThreadRunningTime=600

Indicates that a warning is printed when a thread has been running for 600 seconds.

EventThreshold.ThreadWaitingTime=20

Indicates that a warning is printed when a thread is blocked by another thread for 20 seconds.

For more information, see [Parameters in the tm1s.cfg File](#).

If `EventLogging=F`, the TM1 server does not modify the `tm1s-logs.properties` file.

Note: Warnings aren't printed for the same event unless the thread in the event changed its state.

Because the `ops.logger` relies on enabling a new log file, a default `tm1s-logs.properties` file is created if one doesn't exist. If a `tm1s-logs.properties` file exists, then the new entry is appended and the old one backed up as a copy.

Note: The default file size is 100 MB. The file is overwritten with new data when the limit is reached.

The TM1 Top log

Added v2.0.7 TM1 Server dynamically monitors the threads that are running and outputs thread status to the `tm1top.log` file by using the Top logger. You can also log information about the sandboxes and job queue for a specific database.

You can use the Top logger in IBM Planning Analytics on cloud and IBM Planning Analytics Local. You can download the logs from IBM Planning Analytics Administration. For more information, see [Download database log files](#).

Note: The content of the `tm1top.log` file is the same as the TM1Top logs. The TM1Top utility is scheduled for deprecation on Planning Analytics Local.

Configure tm1s.cfg parameters

The Top logger is Off by default. To enable logging, you must add **TopLogging=T** to the tm1s.cfg file before you start your TM1 Server.

Note: This parameter cannot be configured in Planning Analytics Administration.

The following settings can be configured in the tm1s.cfg file:

TopLogging

Enables dynamic logging of the threads running in an instance of the TM1 Server. The Top logger can also display information about the sandboxes and job queue for a specific database.

Boolean, dynamic

Default=FALSE

TopScanFrequency

The logging frequency (interval) in seconds. The smaller the period, the faster the size of the log file increases. Management of the TM1 database thread log file is the same as the management of the TM1 database message log. The amount of concurrent activity in the TM1 model also determines the rate the TM1 database thread log file increase in file size.

Integer, dynamic

Default is 5 seconds

0 disables the Top logger.

TopScanMode.Threads

Enables logging of the current processing state of each thread. This information includes the name of the user or process that started the thread, the API function that the thread is executing, the lock status of the last object that was locked, the number of objects that are used by the thread, and the total time, in seconds, that the current API function or chore process has been processing.

Default = T

TopScanMode.Sandboxes

Enables logging of the active sandboxes for the current server, the total memory that is consumed for all sandboxes by a user, and the number of sandboxes for this user.

Default = F

TopScanMode.SandboxQueueMetrics

Enables logging of sandbox queue metrics. The name of the node for the sandbox, the status of the sandbox in the queue, and the length of time the sandbox was in the queue before it was processed is logged for each sandbox in the queue.

Default = F

For more information, see [Parameters in the tm1s.cfg file](#).

If TM1 Server starts with **TopLogging** turned on, the server adds the Top logger configuration to the tm1s-log.properties file if the configuration is not already there. The log level is INFO. The special log format is controlled by the format parameter of the definition, which must be set to TM1Top.

The following configuration information is added:

```
log4j.logger.Top=INFO, S_Top
log4j.appender.S_Top=org.apache.log4j.SharedMemoryAppender
log4j.appender.S_Top.MemorySize=5 MB
log4j.appender.S_Top.MaxFileSize=100 MB
log4j.appender.S_Top.MaxBackupIndex=20
log4j.appender.S_Top.File=tm1top.log
log4j.appender.S_Top.Format=TM1Top
```

After the Top logger is enabled and the configuration settings are added to the tm1s-log.properties, you can disable or enable the Top logger without restarting the TM1 Server.

TM1 REST API support

The Top logger and its settings can be read and modified by using the TMI REST API:

```
GET /api/v1/ActiveConfiguration/Administration/TopLog
PATCH /api/v1/StaticConfiguration/Administration/TopLog
{
  "ScanFrequency": "P0DT00H00M02S"
}
GET /api/v1/Loggers('Top')
PATCH /api/v1/Loggers('Top')
{
  "Level": "Off"
}
```

For more information, see [TopLogSettings](#) and [TopScanMode](#).

Log example

```
==== 2019-01-09 22:20:54 GMT      2019-01-09 17:20:54 local  ==== Server:Planning Sample
=====
10744 Th:Pseudo      -      Idle      -      -
-      -      0
22584 Th:DynamicConfig -      Idle      -      -
-      -      0
676   Admin          -      :1      Idle      -      -
-      -      0
4920  Sa              Architect Idle      -      -
-      -      0
==== 2019-01-09 22:20:56 GMT      2019-01-09 17:20:56 local  ==== Server:Planning Sample
=====
8120  C:SleepEvery10s  -      Run:R     -      Process SyncSleep_10  Prolog
(6)  0
10744 Th:Pseudo      -      Idle      -      -
-      -      0
22584 Th:DynamicConfig -      Idle      -      -
-      -      0
676   Admin          :1      Run:R     -      POST /api/v1/Cubes('plan_BudgetPlan')/Views('def')/
tm1.Execute - 1
22620 > Work unit for 676 -      -      -
-      -      0
3284  > Work unit for 676 -      -      -
-      -      0
22376 > Work unit for 676 -      -      -
-      -      0
17568 > Work unit for 676 -      -      -
-      -      0
2940  > Work unit for 676 -      -      -
-      -      0
128   > Work unit for 676 -      -      -
-      -      0
20552 > Work unit for 676 -      -      -
-      -      0
4920  Sa              Architect Idle      -      -
-      -      0
```

The TM1.Mdx.Interface logger

Added v2.0.7 The TM1.Mdx.Interface logger reports syntax errors when it is set to DEBUG level.

In IBM Planning Analytics version 2.0.6 or earlier, the TM1.Mdx.Interface logger reported syntax errors when it was set to ERROR level. However, when Planning Analytics for Microsoft Excel or Planning Analytics Workspace generated invalid MDX internally, the tm1server log filled up quickly with a large number of entries in the form:

```
11276 [6b5] ERROR 2017-10-16 13:51:10.237 TM1.Mdx.Interface
Syntax error at or near: 'hildren},{[Entity].[]}))}', character position 21
```

These errors now appear only when the logging level is set to DEBUG.

Searching large transaction logs

Added v2.0.7 If there are too many INVALID records, for example more than **500** invalid records in **one search** in the transaction logs, search stops and an error message is generated in the server log. User intervention is required to clean up the transaction logs. This change prevents the server from searching for long periods and blocking other activities.

For example, the error message contains the following text:

```
Starts from <file name>, over 500 bad transaction records found, transaction log searching stopped.
```

The error message indicates where the search was started. For example, if the user searched from December 1 to December 12, the server searches multiple log files. The "Starts from" indicates where in the transaction logs the cleanup must start.

This change in the behavior of searching large transaction logs prevents the TM1 Server from generating large logs. However, user intervention is required to clean up the invalid records if the logs become too large.

Changed v2.0.9 This behavior changed in IBM Planning Analytics version 2.0.9. The default behavior is to search all logs. You can configure a limit to the number of invalid logs that are allowed before search stops. This configuration parameter is private. Contact support to learn more.

Control cubes

Control cubes store minute-by-minute performance statistics for clients, cubes, and servers. This performance data is tracked and stored in the following TM1 control cubes; }StatsByClient, }StatsByCube, }StatsByCubeByClient, and }StatsForServer.

When you enable performance monitoring, TM1 populates several control cubes on a minute-by-minute basis. You can then browse these cubes to analyze the server performance.

The following control cubes are populated during performance monitoring. For more information, see Appendix B, "Control Cubes," on page 121.

Control Cube Name	Description
}StatsByClient	Tracks the message count, average message size, total elapsed time, and other measures for each client on the server.
}StatsByCube	Tracks the memory used for each cube on the server.
}StatsByCubeByClient	Tracks the number and elapsed time of cell updates, cell retrievals, view calculations, and view retrievals for each client and cube on the server.
}StatsForServer	Tracks the connected clients, active threads, and memory used by the server.

Enabling performance monitoring

You can enable performance monitoring during a server session.

Procedure

1. Select the server in the Server Explorer.
2. Click **Server, Start Performance Monitor**.

Results

You have enabled performance monitoring on a per-session basis.

If you want to enable performance monitoring at the start of every server session, you can set the PerformanceMonitorOn parameter in the Tm1s.cfg file to automatically begin performance monitoring when a server starts.

Disabling performance monitoring

You can end performance monitoring during a server session.

Procedure

1. Select the server in the Server Explorer.
2. Click **Server, Stop Performance Monitor**.

Viewing performance statistics for clients, cubes, and servers

After enabling performance monitoring, you can view the status.

Procedure

1. In the Server Explorer, click **View, Display Control Objects**.
All of the Control Cubes will now be displayed, including the Performance Monitoring Control Cubes.
2. Open any of the Performance Monitoring Control Cubes to view the cube.

The available control cubes include:

- StatsByClient
- StatsByCube
- StatsByCubeByClient
- StatsForServer

For more information, see [Appendix B, “Control Cubes,” on page 121](#) and [Appendix C, “Control Dimensions,” on page 135](#).

Note: The performance monitor does not update to reflect any new cubes created or new users added *while* the performance monitor is running. Restart the performance monitor to update it with items that were added.

TM1 performance counters

TM1 performance counters are a collection of continuously updated values that provide real-time monitoring of specific TM1 Server properties and activities such as usage of cubes, views, subsets, dimensions, and read/write activity. Performance counters are viewable in graphical format using the Microsoft Windows Performance Monitor, or in a text-only display using the TM1 PerfMon utility.

These values are incremented and decremented in real time to track specific TM1 properties and activities for cubes, views, subsets, dimensions, and read/write activity.

You can view performance counters with the following tools:

- TM1 PerfMon utility - A text-only display of TM1 performance counters that runs in a console window on Microsoft Windows and UNIX systems. The TM1 PerfMon utility can monitor local TM1 servers only.
- Microsoft Windows Performance Monitor - A Microsoft Windows tool that provides an interactive, graphical display of TM1 performance counters to monitor local and remote Windows TM1 servers.

Important: You need the same administrator rights and privileges to run TM1 PerfMon on a Microsoft Windows system as you do to run the Microsoft Windows Performance Monitor.

- The user must be a member of the local Administrators group to run either TM1 PerfMon or the Microsoft Windows Performance Monitor.
- The user must be a member of the Administrators group, on both the local and remote systems, to monitor a remote TM1 server using the Microsoft Windows Performance Monitor.

Viewing TM1 performance counters with the Microsoft Windows Performance Monitor is not supported on 64-bit Intel Itanium II systems. When running TM1 with this configuration, use the TM1PerfMon console utility to view TM1 performance counters.

Available TM1 performance counters

TM1 provides a set of more than 30 performance counters, organized into groups, such as Cubes, Dimensions, Subsets, Memory, Threads, and Views.

Examples of TM1 performance counters include:

- Threads created
- Views and Dimensions created and destroyed
- Subsets created, duplicated, destroyed, and deleted
- Read lock requests and waits

Viewing TM1 performance counters with the TM1 PerfMon utility

TM1 PerfMon is a console utility for Microsoft Windows and UNIX systems that provides a text display of TM1 performance counters.

Running TM1 PerfMon

Run the `tm1perfmon.exe` file from a command line on either a Microsoft Windows or UNIX system.

Use the following format and parameters:

```
tm1perfmon -servername Name -loop LoopCount -sleep SleepTime
```

Where:

- *Name* is the name of the TM1 Server to monitor. Use quotes when the server name includes spaces. For example:

```
-servername "planning sample"
```

- *LoopCount* is the number of times to repeat, or refresh, the display of counter values. Default value is 1, no loop. This value is optional.

A value of 0 causes TM1 PerfMon to loop indefinitely.

To cancel TM1 PerfMon while looping, press the CTRL+Break keys to return to the command prompt.

- *SleepTime* is the time, in seconds, between loops. Default value is 5 seconds. This value is optional.

For example, the following command line would run TM1 PerfMon continuously to monitor a TM1 Server named `planning sample`. The display would continue to refresh every 5 seconds, using the default value for the sleep option, until you exit by pressing the CTRL+Break keys.

```
tm1perfmon -servername "planning sample" -loop 0
```

The following command line would run TM1 PerfMon to monitor the `salesdata` TM1 Server, looping a total of 10 times with a 5 second refresh rate between display updates.

```
tm1perfmon -servername salesdata -loop 10 -sleep 5
```

If you do not include values for the loop and sleep options, as shown in the following example, TM1 PerfMon will run and display the TM1 performance counters once only.

```
tm1perfmon -servername salesdata
```

Note: You can turn updates to performance counters on or off for a TM1 Server by setting `PerfMonIsActive` in the `tm1s.cfg` file. See [PerfMonIsActive](#).

You can exit the TM1 PerfMon while the utility is still looping. Press CTRL+Break at any time.

Viewing the TM1 PerfMon display output

You can view display output from TM1 PerfMon.

It looks similar to the following:

```
C:\>tm1perfmon -server cdata
TM1 Version: 9.0.3.226
Output data and time Tue Dec 05 17:36:42 2006
Counter Name/ Value
Threads: Threads created          6
Threads: Threads Creation Retry Attempts    0
Threads: Thread Creations Failures    0
Views: Views Created             23
Views: Empty Views Created        0
Views: Views Destroyed           0
Dimensions: Dimensions Created      0
Dimensions: Dimensions Destroyed    119
Cubes: Cubes Invalidated          0
Cubes: Dependent Cubes Invalidated    1
Subsets: Subset Create Empty        0
Subsets: Subset Calculated by Expression  0
Subsets: Subset Calculated with Empty Expression  0
Subsets: Subset Calculated by Expression  0
Subsets: Subset Created by Expression    0
Subsets: Subset Created by MDX Expression  0
Subsets: Subset Duplicated          0
Subsets: Subsets destroyed          0
Subsets: Subsets Deleted           0
```

The title fields in the display include:

Field	Description
Counter Name	Group name and counter name for each TM1 performance counter. Displayed in the format: GroupName:CounterName For example: Threads:Threads Created
Value	Numeric value for the TM1 performance counter at the specific output date and time.

Viewing performance counters with the Microsoft Windows Performance Monitor

Use the Microsoft Windows Performance Monitor to view TM1 performance counters in a graphical-type display for Windows TM1 servers only.

The Microsoft Windows Performance Monitor is supplied with current versions of Microsoft Windows and runs as an add-in to the Microsoft Management Console.

Opening the Microsoft Windows Performance Monitor

Open the Microsoft Windows Performance Monitor.

Procedure

Click **Start, Settings, Control Panel, Administrative Tools, Performance**.

The Microsoft Windows Performance Monitor opens.

Adding and Viewing TM1 performance counters in the Microsoft Windows Performance Monitor

You can view TM1 performance counters by adding them with the Add Counters dialog box in the Microsoft Windows Performance Monitor. This dialog also lets you view a brief explanation of each counter.

This dialog also lets you view a brief explanation of each counter.

Procedure

1. Click **Add** in the Performance Monitor toolbar.

The Add Counters dialog box displays.

Note: Microsoft Windows can sometimes display the following message when you open or close the Add Counters dialog.

"At least one data sample is missing. Data collection is taking longer than expected. You might avoid this message by increasing the sample interval. This message will not be shown again during this session."

This message is informational only and typical if the system becomes too busy when adding new counters. Click **OK** to clear the message.

2. Select **IBM Cognos TM1** from the Performance object list.

The available performance counters display in the performance counter list.

3. Select a performance counter from the list of available counters.

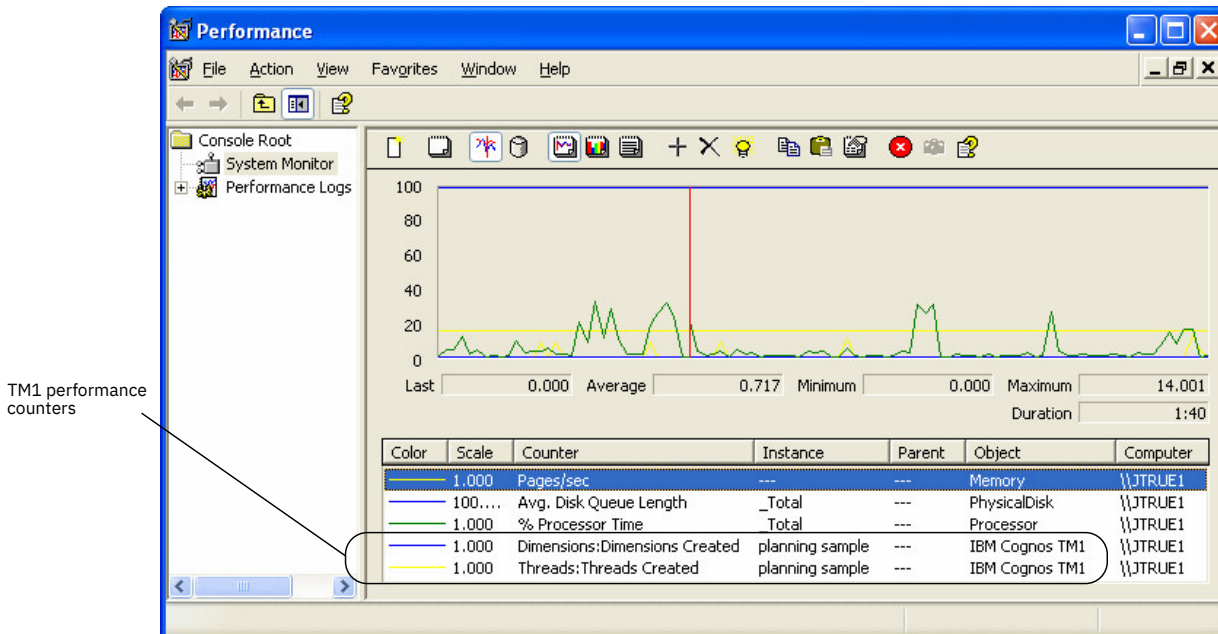
You can also click Explain to see a text description of the currently selected counter.

4. Select the instance of the server you want to monitor.

5. Click **Add** to add the selected counter to the performance display.

6. Click **Close** to close the Add Counters dialog and return to the Performance Monitor.

The Performance Monitor will add the TM1 performance counters to the monitoring process and update the display.



Chapter 7. IBM Planning Analytics tools and utilities

Use the following tools and utilities with IBM Planning Analytics.

Cognos TM1 Operations Console Deprecated in v2.0.9

Web-based advanced graphical user interface. See [“IBM Cognos TM1 Operations Console workflow” on page 66.](#)

Deprecated in v2.0.9 Performance Management Hub

A server framework that provides common application services to IBM Planning Analytics for Microsoft Excel, Operations Console, and other TM1 client applications. You can set configuration options using the web interface. See [“IBM Cognos Performance Management Hub” on page 80.](#)

tm1xfer

Command line utility used to compress TM1 Server database files and move from one platform to another platform while preserving mixed case names for objects on both Microsoft Windows and UNIX platforms. See [“tm1xfer” on page 94.](#)

ODBC_test

The `odbc_test` tool is used to diagnose and test the TM1 ODBC connection on UNIX. See [“odbc_test tool” on page 101.](#)

Debugging tools

For tools designed specifically for debugging TM1 Server, see [“Other debugging tools” on page 90.](#)

TM1RunTI

Command line interface tool that initiates a Cognos TM1 TurboIntegrator (TI) process from within any application capable of issuing operating system commands. See [Using TM1RunTI](#) in the *TM1 TurboIntegrator* documentation.

Synchronized()

Cognos TM1 TurboIntegrator (TI) function used in a TurboIntegrator script to force serial execution of a designated set of TurboIntegrator processes. See [Serializing TurboIntegrator processes using synchronized\(\)](#) in the *TM1 TurboIntegrator* documentation.

Using the IBM Cognos TM1 Operations Console

Deprecated in v2.0.9 The TM1 Operations Console is a web-based utility that displays and logs TM1 server activity. You can use TM1 Operations Console to set up and view log files.

Note:

IBM TM1 Operations Console is deprecated in IBM Planning Analytics version 2.0.9. You can use IBM Planning Analytics Administration to monitor databases in Planning Analytics and Planning Analytics Local.

To learn more, see [Monitor and administer databases.](#)

- *Transactions Log*

A log file that tracks the data transactions made by the clients logged in to one TM1 Server. Recorded information includes the date and time the change was made, name of the client who made the change, before and after value, name of the cube in which the change was made, and the elements that identify the cell that changed.

- *Server Message Log*

A log file containing details on the activity of the TM1 Server, such as an executed processes, chores, loaded cubes and dimensions, and a synchronized replication.

- *Audit Log*

A log file that monitors changes to TM1 objects and system-wide events, such as modifications to dimensions, views and subsets and successful/ unsuccessful login activity.

IBM Cognos TM1 Operations Console workflow

Deprecated in v2.0.9 Use this list to identify the tasks you need to do in the Cognos TM1 Operations Console and where to find more information about them.

Note: Cognos TM1 Operations Console is installed by default when you install TM1. For more information, see [Cognos TM1 Operations Console installation](#).

By default, the Cognos TM1 Operations Console is configured to monitor the **SData** TM1 server with a monitoring group called **Admin**. To change those defaults, use the [pmhub configuration](#).

1. Create your [logical operations group](#).
2. [Verify admin access](#) on that server.
3. For security purposes, you can “[Display user names](#)” on page 68 .
4. The Health Status Check shows the current status of the server at a glance with icons. See “[Monitoring in the Cognos TM1 Operations Console](#)” on page 69 for the list.
5. To use the default set of server configuration parameters, skip to the next step. If you want to change the basic parameters, see [Configure](#).
6. To get a quick status report, double-click the server name to create a new Health Status tab as described in “[Server Health Status](#)” on page 68. The Health Status tab is populated with the Enhanced version of the status report. You can add tabs and populate those tabs with the other kinds of log files.
7. Click the **Configuration mode** icon so you can configure other log files.
8. Click the server selection check box. This action reveals the **Schedule new log** buttons. You can drag the bottom pane lower to reveal the full contents of the window. Click the **Schedule new log** button and define the parameters for this new log.
9. To see the log, switch to Monitor mode and right-click the server and select **View Log**. This action opens a new Monitors tab and populates it with the scheduled logs. Nothing displays if the current time is earlier than the start time. You can verify admin access as well. You can [Filter](#) the results to make it more readable.
10. After set up, you can [save and reload](#) a log file configuration.
11. You can change the [layout](#) to suit your needs. You can also repopulate the bottom pane with other log information.
12. Set a [Watchdog](#).
13. You can also monitor the TM1 Applications server in the TM1 Operations Console, as described in “[Monitoring the Cognos TM1 Application Server](#)” on page 72.

Configuring the Cognos TM1 Operations Console

Deprecated in v2.0.9 By default the Cognos TM1 Operations Console is configured to use a sample adminhost of localhost, server of sdata, and a group called admin for authentication so you can get up and running quickly. If you prefer, you can edit these configurations to customize the authentication server.

Procedure

1. Enter `http://servername:port number/pmhub/pm/admin` to open the configuration screen.
2. To change the default adminhost, server, and group for monitoring, expand **Configurations > Operations Console TM1 Monitors** . Click in each setting to change it.

If you want to use Microsoft Internet Explorer 8 with the Cognos TM1 Operations Console see the Microsoft Internet Explorer documentation and ensure these settings:

- Enable the option to refresh web pages with every visit
- Disable the options for script debugging

Starting and logging into the Cognos TM1 Operations Console

Deprecated in v2.0.9 To open the IBM Cognos TM1 Operations Console, enter the URL that identifies the port and server name for the component.

Procedure

1. In a web browser, type the following web address: `http://servername:port_number/pmhub/pm/opsconsole`

where

- *servername* is the computer where the Cognos TM1 Operations Console and your web application server are installed. You can use the keyword `localhost` if you are currently logged on to the Web server that is running Cognos TM1 Application Web. Or you can use the machine name, domain name, or IP address of the Web server hosting the application.
- *port_number* is the port number where your web application server is running. For the version of WebSphere® Liberty that is provided with the Cognos TM1 installation, the default port number is 9510.

2. On the log in page, enter values for the following fields, and then click **Log In**.

To use the default monitoring group, enter



- **Namespace:** Use the pull-down to select the available Namespace. If the system is a CAM-secured system, the namespace is the Cognos Analytics namespace that the TM1 system is secured against. If the system isn't CAM secured the namespace is *adminhost/tm1 server* for example **localhost/SData**
- **User Name:** admin
- **Password:** apple

See [Setting up the Cognos TM1 Operations Console](#) to get started.

Setting up the Cognos TM1 Operations Console

Deprecated in v2.0.9 IBM Cognos TM1 TM1 Operations Console setup tasks.



Before you can use the TM1 Operations Console, add a production group and identify the servers you want to monitor.

The TM1 Operations Console opens in Monitor mode. You can click the Configuration  icon to switch to Configuration mode to configure the logs and watchdog reports. Use the Monitor  icon to switch back to monitor mode at any time.

Adding an Operation Group, Adminhost, and servers

Deprecated in v2.0.9 Once the IBM Cognos TM1 Operations Console is installed and running, add an Operation Group and identify the adminhost and servers.

For example, you might have a group of servers designated as "Development," "Production," or "Test."

1. Click **Add operation group**  from the toolbar or use the Action menu  to select **Add Operation Group**.
2. Enter a name for the group. You can use any name here.
3. Click **Create**.
4. Right-click the operation group and select **Add Adminhost**.
5. Enter the fully-qualified name of the AdminHost. The default setup uses `localhost`.
6. Click **OK**
7. Right-click the AdminHost you just added and select **Add TM1 Server**.
8. Enter the name of the server you want to monitor. To use the default setup, enter **SData**.

See [Monitoring the Cognos TM1 Applications Server](#) for details on how to monitor a TM1 Application server in the TM1 Operations Console.

Verify admin access

Deprecated in v2.0.9 Before you can perform certain functions in the Cognos TM1 Operations Console, you must enter valid administrator credentials.

Procedure

1. Right-click a TM1 Server.
2. Select **Verify admin access**.
3. Enter valid username and password credentials for an administrator ID.
4. Click **OK**.

Display user names

Deprecated in v2.0.9 The `MaskUserNameInServerTools` parameter determines whether user names are displayed or masked out in the IBM Cognos TM1 Operations Console.








When `MaskUserNameInServerTools=TRUE` is set in the `Tm1s.cfg`, user names are masked in the Cognos TM1 Operations Console for security purposes. To unmask the names, the administrator can right-click the server in the Cognos TM1 Operations Console and then click **Verify admin access** to verify the administrator status.

This parameter is set to `FALSE` by default. If you do not explicitly set this parameter to `TRUE`, the Cognos TM1 Operations Console displays user names even if administrator access was not verified.

Server Health Status

Deprecated in v2.0.9 The Server Health Status tab uses icons to provide almost instant feedback on the status of the servers monitored by the Cognos TM1 Operations Console.

Note: You must [verify admin access](#) before data can display.

Icon	Status
	Online. Server is online. This server is running.
	Starting. The server is starting.
	Entering Bulk Load Mode. The Server is entering Bulk Load Mode.
	Bulk Load Mode. The server is in Bulk Load Mode.
	Offline. Server is offline/not able to contact. This server is not running or is unable to connect to the Cognos TM1 Operations Console.
	Stopping. The server is stopping.
	Unsupported. Heartbeat not supported.

The Health Status tab also displays the number of threads in the server and the number of threads that are waiting for a lock to be granted.

Since the Health Status is updated every 10 seconds, the information is current to within 10 seconds or less depending on when the server status changed since the last refresh

If the report contains a refresh rate, the data is updated at the refresh rate specified for that report.

Monitoring in the Cognos TM1 Operations Console

Deprecated in v2.0.9 The following reports and logs are available in the IBM Cognos TM1 TM1 Operations Console.

To see these reports, right-click the server you want to use, and select **Monitor** and the report you want. Before data can display, you may need to [verify admin access](#) or [schedule a log](#) if one is not being automatically scheduled.

You can also choose to save the [log to disk](#) or [export it](#). See [Filtering](#) for details on how to reduce the displays.

For some of these reports, drag the lower pane down to reveal the full content of the window.

Status

Basic or Enhanced



CAUTION: Using the default Enhanced version of this report when the server is under heavy load can degrade system performance by up to 10%.

Sandbox

Shows the threads being used in sandboxes on the server.

Sandbox Queue

Shows the jobs in the sandbox queue.

Performance Statistics

Shows the server memory and performance statistics report. You must turn the gathering of performance statistics by right-clicking the server name and selecting **Start Performance Monitor** before this report can display information.

Processes

Shows the thread involved in any TurboIntegrator processes.

Chores

Shows server activity involved with Chores.

Memory Usage Graph

Provides a graphical display of memory usage on the server.

Thread Details Graph

Provides a graphical display of thread activity on the server.

You can use these logs:

Transaction log file

Displays the transactions recorded in Tm1s.log when a TM1 client changes a cube value.

Message log

Displays the TM1 server records status messages on the activity of the server in a log file. These messages contain details on activity such as executed processes, chores, loaded cubes and dimensions, and synchronized replication.

Audit log

Displays changes to metadata, such as modifications to dimensions, views and subsets.

Basic and enhanced server status reports

Deprecated in v2.0.9 You can view basic and enhanced versions of the status information for a server.

About this task

Double-click the server name or right-click the server and select **Monitor** > **Status** > **Enhanced** or choose **Basic**.



CAUTION: Using the default Enhanced version of this report when the server is under heavy load can degrade system performance by up to 10%.

Results

The report displays in a new Monitor tab. To reduce the content, use the **Filter** button (see “[Filtering results in the IBM Cognos TM1 Operations Console](#)” on page 76). To export the log to a file, select **Log to Disk**.

Sandbox and sandbox queue reports

Deprecated in v2.0.9 Sandbox activity

Procedure

Double-click the server name or right-click the server and select **Monitor > Sandbox** or **Sandbox queue**.

Results

The report displays the sandbox activity on the server in the Monitor tab. You will only see data here when sandboxes are turned on and being used.

Performance Statistics reports

Deprecated in v2.0.9 You can monitor the server memory and performance statistics for a server in the Cognos TM1 Operations Console.

Procedure

1. To begin gathering server and performance statistics, right-click the server and select **Start Performance Monitor**.
2. To see the log, right-click the server and select **Monitor > Performance Statistics**.

The following statistics are available on the Performance Statistics report:

- Memory Used for Views
- Memory Used for Calculations
- Memory Used for Feeders
- Memory Used for Input Data
- Total Memory Used

In addition, the Status options for each server being monitor displays the following statistics:

- MemoryUsed
- GarbageMemory

TurboIntegrator Processes reports

Deprecated in v2.0.9 You can monitor the TurboIntegrator Processes for a server in the Cognos TM1 Operations Console.

Procedure

1. If you have not started Performance Monitoring, right-click the server, verify admin access, then right-click the server and click **Start Performance Monitor**.

If Performance Monitoring has not been turned on, default values or the values obtained from the last run display.

2. Right-click the server and select **Monitor > Processes**

The following statistics are available on the Process pane: Process Name; Current[®] State; Completion Status; Client Name; Last Start Time; Last End Time; Last Duration. Client Name contains information about the threads associated with a particular process. Each entry in the client name field is in the format *Client ID Thread ID*. Multiple entries are separated by a semi-colon.

If the thread originated by a client, the name includes the client name, such as Admin, 512 where admin is the client name and 512 is the corresponding thread ID.

If the thread is originated by a chore, the chore name appears in square brackets along with its corresponding thread ID.

When a large number of threads are associated with a process click on the node arrow to provide a list such as:

```
Completion Status :  
NORMAL - Process executed normally  
UNKNOWN - Process status could not be captured  
REJECTED_RECORD - User asked to skip this record and note an  
error into the log with the ItemReject() TI function  
MINOR_ERRORS - Execution finished but had less than  
the maximum number of minor errors  
PROCESS_BREAK - User executed a ProcessBreak() function  
ABORTED_BY_PROCESSQUIT - Execution was aborted by a  
ProcessQuit() TI function  
ABORTED_WITH_ERROR - Process was aborted because a  
serious error occurred  
ABORTED_ON_INIT - Process was aborted because of a  
serious error during the process startup phase.
```

The Process Pane is refreshed based on the rate specified by the Memory Stats Refresh Rate in the Configure window.

Monitoring Chores

Deprecated in v2.0.9 You can monitor the Chores for a serving in the Cognos TM1 Operations Console

Procedure

1. If you have not started Performance Monitoring, right-click the server, verify admin access, then right-click the server and click **Start Performance Monitor**.
2. To see the statistics for the Chores running on a server, right-click the server and click **Monitor > Chores**

The following information is shown for chores: Chore Name, Completion Status, Current State, Client Name, Last Start Time, Last Duration in seconds, Next Activation Time, Current Process.

Memory Usage Graph

Deprecated in v2.0.9 Displays memory usage as a graphical chart.

Right-click the server you want to use and select **Monitors > Memory Usage Graph**.

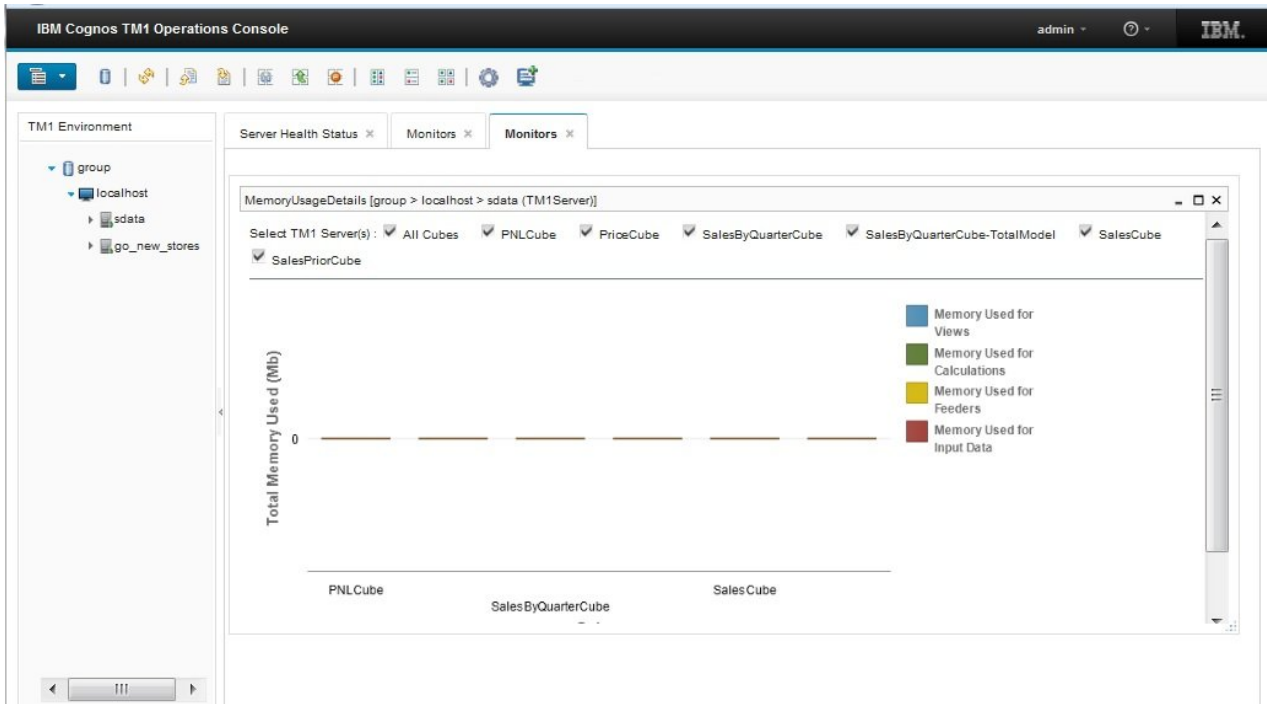


Figure 1. Memory Usage Graph

Thread Details Graph

Deprecated in v2.0.9 The Thread Details Graph displays thread information as a graphical chart.

Right-click the server you want to use and select **Monitors > Thread Details Graph**.

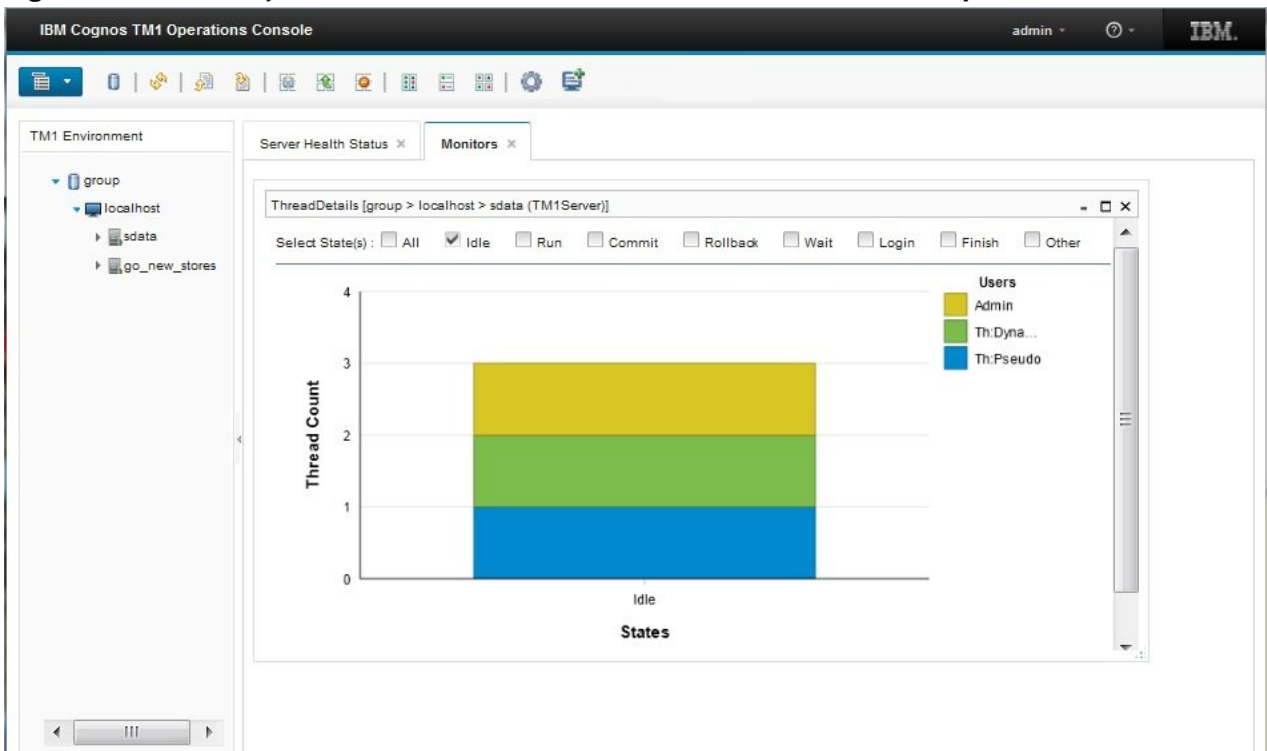


Figure 2. Thread Details Graph

Monitoring the Cognos TM1 Application Server

Deprecated in v2.0.9 You can monitor the server activity of a Cognos TM1 Application Server.

Procedure

1. To add a Cognos TM1 Application server to the Cognos TM1 Operations Console, right-click the operation group where you want to list the Cognos TM1 Application Server.
2. Select **Add Application Server**.
3. Enter a name to use to identify this Application Server and click **Create**.
4. Right-click the name that you just added and select **Configure**.
5. Complete the fields on that dialog box:

IP

Enter the IP address for the Applications Server. The IP field can also be the fully-qualified domain name or the NetBIOS name.

Context

Enter the name of the planning service, for example, pmpsvc

Port number:

Enter the port number specified in the JVM for TM1. This port number is the jmx port number that you have specified in the applications server JRE options.

Top Refresh period (sec)

A typical refresh period is 2.

Top Tolerance Factor (sec)

Enter 0 unless you want to expand the tolerance.

Top Time Out (sec)

A typical time out value is 1.

6. Click **OK**. If the Cognos TM1 Applications Server is running, the health status becomes green and the activity is being monitored. If the Cognos TM1 Application Server status is not green, ensure that the service is running using IBM Cognos Configuration.
7. Most users prefer to use SSL to securely monitor the TM1 Applications Server. However, you can also monitor it without using SSL. This is a less secure method but does not require the additional certificate steps described in [“Using TLS when monitoring the TM1 Applications Server”](#) on page 73. To monitor the TM1 Application Server without using SSL, modify the jvm parameter

```
-Dcom.sun.management.jmxremote.ssl=true
```

and change it to

```
-Dcom.sun.management.jmxremote.ssl=false
```

8. Restart the TM1 Applications Server.

Using TLS when monitoring the TM1 Applications Server

Deprecated in v2.0.9 When you use TLS, take these additional steps to monitor the Applications Server.

To use TLS, you need to create a security certificate on the machine where the Cognos TM1 Application Server is running, then export that certificate to the Cognos TM1 Operations Console machine and configure the Cognos TM1 Operations Console machine so that it uses that new certificate store.

See [Configure the TM1 Server to use custom TLS](#) in *Planning Analytics Local Installation and Configuration* for more information.

Log files for the TM1 Applications server with the TM1 Operations Console

Deprecated in v2.0.9 The IBM Cognos TM1 Operations Console can track and monitor activity in the TM1 Applications server..

When you select **Log To Disk** for automatic logging, the rate at which log messages are written is the same as the refresh rate of data in the monitoring window. With **Log to Disk** for Cognos TM1 Applications Server monitoring, the older monitoring data from the log file is overwritten with new log contents.

Unlike configuration of a Cognos TM1 server, you cannot use **Log Append** with the Cognos TM1 Application server.

Cognos TM1 Application Server logging can also be scheduled. Two types of Application Server data can be logged:

- Application Server Statistics
- Application Server Session Information

The **Log to Disk** option is available on all monitor windows. Click **Log to Disk** to create an automatic log. The data will continue to be logged until you turn off the logging by unchecking **Log to Disk**, closing the monitor window, or closing the Cognos TM1 Operations Console.

Log files in the IBM Cognos TM1 Operations Console

Deprecated in v2.0.9 You can schedule a log, use the log to disk option, or enable automatic logging.

Before any data can display in a log, you must [“Verify admin access”](#) on page 68.

You can access these [TM1 server logs](#) using the TM1 Operations Console:

Transaction log file

Displays the transactions recorded in Tm1s.log when a TM1 client changes a cube value.

Message log

Displays the TM1 server records status messages on the activity of the server in a log file. These messages contain details on activity such as executed processes, chores, loaded cubes and dimensions, and synchronized replication.

Audit log

Displays changes to metadata, such as modifications to dimensions, views and subsets.

Right-click the server and select **View Log**, **Message Log**, **Transaction Log**, or **Audit Log** to open these log files. See the "System and Performance Monitoring" chapter of the *TM1 Operations* documentation for details on these logs and how to enable them.

Working with logs

Deprecated in v2.0.9 These topics describe how you can adjust the display of log files.

Configuring the log file and server numeric parameters

Deprecated in v2.0.9 Use the **Configure** option to set the numeric parameters that define how the log file numerics and other server parameters are gathered.

Complete the following parameters for each log file.

Log period

Specifies the time interval between updates being written to the log file.

For example, if the screen Refresh is set to 2 seconds, LogPeriod could be set to 10 seconds so that every fifth screen display will be output to decrease the amount of data written to the file.

Default is 2 seconds.

Log Append

By default a new log file is not appended, it overwrites the existing log. Select True to append new logfiles to the existing log.

Refresh period

By default, the log waits 2 seconds before refreshing the data. You can enter a longer or shorter time for the refresh here in seconds.

Tolerance factor

By default, the data is new (0). To permit data in the log to be older, enter a number here in seconds for how old the data can be before requiring a new polling.

Time Out

By default, attempts to connect to the server end after 2 seconds. You can change this number to shorten or lengthen the time the log will keep trying before giving up.

If the IBM Cognos TM1 Operations Console server does not get the updated status in the seconds specified here, then the old data (if it exists) is sent with an indication that a timeout has occurred.

Memory Stats refresh period

By default, the log waits 2 minutes before refreshing the data. You can enter a longer or shorter time for the refresh here in minutes.

Memory Stats tolerance factor

By default, the data is new (0). To permit data in the memory statistics to be older, enter a number here in seconds for how old the data can be before requiring a new polling.

Memory Stats timeout

By default, attempts to connect to the server end after 2 minutes. You can change this number to shorten or lengthen the time the log will keep trying before giving up.

If the IBM Cognos TM1 Operations Console server does not get the updated status in the minutes specified here, then the old data (if it exists) is sent with an indication that a timeout has occurred.

Scheduling logs

Deprecated in v2.0.9 You can schedule a log, use the log to disk option, or enable automatic logging.

Procedure

1. To schedule a log, click the **Configuration > Logging** tab.
2. Depending on the kind of server you want to monitor, click either the **TM1 Servers** tab or the **TM1 Applications** tab.
3. Check the server you want to create the log for.

You may need to drag the pane to reveal the schedule log icons.

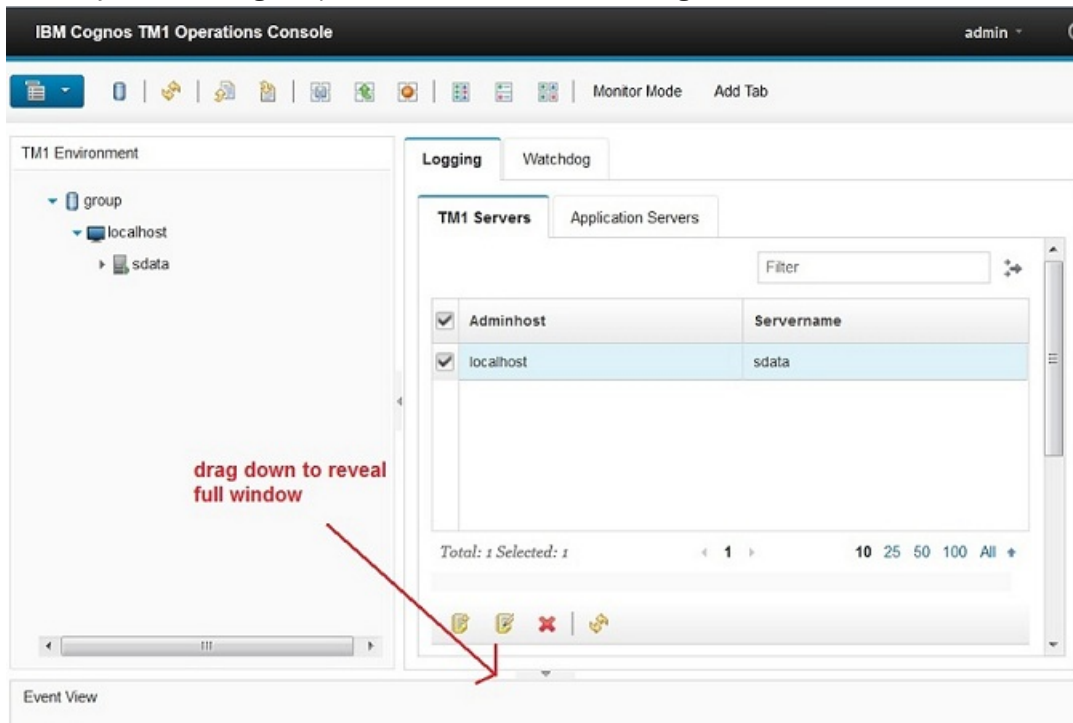



Figure 3. Revealing the bottom part of the TM1 Operations Console pane.

4. To create a new log, click **Schedule New Log** .
5. Define the parameters of the log:

Log type

Choose one or more kinds of logs: **Log Status**, **Log Enhanced Status** (includes object contention columns), **Log Sandbox**, **Log Sandbox Queue**

Log Duration

Enter the **Start Date**, **Start Time**, **Stop Date**, **Stop Time**, and **Log Frequency** in seconds. Click in the fields to open a calendar to set the dates.

States


Check the states to log: **Idle**, **Run**, **Commit**, **Rollback**, **Wait**, **Login**, or **Finish**.

Threads

Check the type of threads to log: **System Threads**, **Chores**, or **User Threads**.

6. When the details are defined, click **Create**.

7. You can use the **Filter** button to reduce the number of logs shown on the screen.

8. Use  to change the parameters of the log file after it is saved.

Using Log to Disk to save log files

Deprecated in v2.0.9 Log files can be stored using the Log to Disk checkbox that displays on most log screens.

Procedure

1. Click the **Log to disk** option found on most logs.

Log to disk generates a log of the current screen activity.

2. To see the log, you can right-click the server and select **View Log**. Check the log you want to see and click **OK**.

3. Or go to the log location for that server, for example **C:\Program Files\IBM\cognos\tm1_64\bin64\opsconsoledata\localhost\sdata\admin\Logs** to find the .csv file of the log.

Viewing logs

Deprecated in v2.0.9 After you have generated a log, use the View Log option to display it.

A user monitoring a server can see all the schedules created by other users. The user can also perform all supported actions on those schedules. However, if the owner deletes the monitored server, that schedule is deleted and becomes unusable by others.

Procedure

1. Right-click the server whose logs you want to view.

2. Select **View log**.


Results

The View logs dialog box displays with the list of all logs. The Log Type indicates if the log is scheduled or automatic. You can use the **Filter** button to restrict the display to only results you are interested in.

Filtering results in the IBM Cognos TM1 Operations Console

Deprecated in v2.0.9 You can reduce the number of entries found in a log or list using filtering.

Filtering logs

On the Monitor tab, you can filter the results. When no filter is applied to the results, the  is displayed. Click this button to display the list of states you can use to filter the results: Idle, Run, Commit, Rollback, Wait, Login, Finish. Next, select the threads you want to monitor: System Threads, Chores, User Threads.

Filtering lists

Use the **Filter** field and **Apply Filter** buttons to filter a list of servers or logs. You can type ahead in this field to identify any aspect of the displayed servers or logs.

For example, on the Logging tab you can filter the list of servers.



Figure 4. Applying filters

Exporting logs

Deprecated in v2.0.9 You can export a log file.

Procedure

1. Click the **Export** option found on most logs.
2. Select the log content from the window that displays.
3. Paste the content into another file, such as a text file.

Download a csv version of the log file

Deprecated in v2.0.9 To download a csv version of the log, use the **Download Log File** option.

Procedure


1. Right-click the server you want and select **Download Log File**.
2. Click the selection button for the log you want to download.
Anytime you choose the **Log_to_Disk** option on a log it creates an entry here for the current version of the screen.
3. Confirm that you want to download the log and complete the Save or Open dialog box as needed.
4. Browse to the location to store this file and click **Save**.

Downloading and Uploading the Cognos TM1 Operations Console configuration file

Deprecated in v2.0.9 You can save the details of the Cognos TM1 Operations Console setup using the Upload and Download Configuration File options.

Procedure

1. To save the current configuration for the Cognos TM1 Operations Console, click **Download**

Configuration File  and save the file to the location of your choice.

2. To install the current `tm1opsconsoleconfig.xml` configuration, click **Upload Configuration File**



, browse to the location where of the configuration file and click **OK**

TM1 server Transaction log

Deprecated in v2.0.9 The transaction log lists transactions occurring in the specified server over the specified time frame. See the "System and Performance Monitoring" chapter of the *TM1 Operations* documentation for details on this log and how to use it.

Procedure

1. To get a log of transaction activity, right-click the server and select **Transaction log**.
2. Click the Start date; Start time; End Date; End time to define the time period over which transactions will be logged.
If you set just Start time and End time, the time period defaults to all day.
3. You can also use the User, Cube, or Flag filter to identify specific values in each of these columns that you want to use as a filter.
4. To generate a txt file of the log, click **Export** and select the resulting window's contents. Then you can paste that information into a blank text file.
5. To restore any changed data, click **Back Out**.
6. When the parameters are set, click **OK** to retrieve data from the server to the TM1 Operations Console. Logging continues until you log out of the session or the end time period is met.

TM1 server Message log

Deprecated in v2.0.9 Displays the TM1 server records status messages on the activity of the server in a log file. These messages contain details on activity such as executed processes, chores, loaded cubes and dimensions, and synchronized replication. See the "System and Performance Monitoring" chapter of the *TM1 Operations* documentation for details on this log and how to use it.

Procedure

1. To get a log of messages, right-click the server and select **Message log**.
2. Click the Start date; Start time; End Date; End time to define the time period over which transactions will be logged.
If you set just Start time and End time, the time period defaults to all day.
3. You can also use the Thread, Level, or Logger filter to identify specific values in each of these columns that you want to use as a filter.
4. To generate a txt file of the log, click **Export** and select the resulting window's contents. Then you can paste that information into a blank text file.
5. You can also use **View Process log** to see the process log.
6. When the parameters are set, click **OK** to initiate logging. Logging continues until you log out of the session ends or the end time period is met.

TM1 server Audit log

Deprecated in v2.0.9 Displays the TM1 server audit log on the activity of the server in a log file. See the "System and Performance Monitoring" chapter of the *TM1 Operations* documentation for details on this log and how to use it.

Procedure

1. To get a log of the audit activity, right-click the server and select **View Audit log**.
2. Click the Start Date; Start Time; End Date; End time to define the time period over which transactions will be logged.
If you set just Start date and End date, the time period defaults to all day.

3. You can also use the Object Type; Object Name; Owner Type; Owner Name; Event Type; Event Code to identify specific values in each of these columns that you want to use as a filter.
4. To generate a txt file of the log, click **Export** and select the resulting window's contents. Then you can paste that information into a blank text file.
5. You can also use **View Detail** to see individual log entries.
6. When the parameters are set, click **OK** to initiate logging.
Logging continues until you log out of the session ends or the end time period is met.






Using Watchdog to monitor server activity

Deprecated in v2.0.9 You can use Watchdog rules to establish a set of criteria that identifies specific states in the servers or processes being monitored and the action to take when the criteria are met. For example, you could create a Watchdog rule that writes to the log when the number of threads reaches 50 on an IBM Cognos TM1 server.

About this task

For a TM1 server, only one Watchdog can be configured by any user having server administrator credentials. It is then available for use by other users. If a user who sets up a server deletes the monitored server, the Watchdog running on that server is deleted and the Watchdog stops running.

Procedure

1. Click .
2. Click the **Watchdog** tab.
The tab shows the servers being monitored by the user and the states of the Watchdog rules that have been set up for the servers.
3. Click the server you want to add the Watchdog rule to.
4. To edit the details of the Watchdog rules, you must first verify your admin status. Click **Verify** and enter the username and password for the administrator user on this server.
Remember: You may need to drag the pane to expose the rules setup area in the bottom pane.
5. To create a new rule, click .
6. Use the grid to specify when you want a Watchdog event to take place. Click the **Field** to use for the criteria, such as state; the **Operator** to apply, such as Equals; and a value.
For example, you can set State equals Busy. You can use any appropriate value for the fields. The State field accepts these values: Run, Idle, Busy, Commit, Rollback, Logon, Wait, Finish.
7. By default the **Action** to take when a criteria is met is set to **Kill**. You can change the action to **Log** to write the message to the file.
8. After defining the criteria for the rule, click **Save**.
9. Add other rules, if needed. Use these icons to work with your rules:
 - Click **Edit Rule**  to change the criteria of an existing rule.
 - Click **Delete Rule**  to remove the rule.
 - Click **Refresh**  to refresh the list of rules.
10. Set the **Frequency** for the Watchdog process to run.
11. Click **Save** to save the Watchdog for this server.
12. To put the Watchdog rule into effect, click **Start**.
The Watchdog runs based on the Frequency you entered. If the server meets the criteria of the rule, the action is taken.
13. To stop the Watchdog from running, click **Stop**.

Example

This example logs an event if the thread count on the TM1 server goes above 20:

```
Threadcount Greater Than 20  
Action Equals Log
```

This example stops any process started by the Admin user after 200 seconds:

```
User Equals Admin  
Threshold Equals 200  
Action Equals Kill
```

This example logs an event if any user other than Admin has been running the TurboIntegrator script called Rogue for over 100 seconds:

```
User Not Equals Admin  
Threshold Equals 100  
Object Equals Rogue  
Action Equals Log
```

This example stops a process called Process_MonthlyDataLoad if it was started by a user other than Admin.

```
User Not Equals Admin  
Object Equals Process_MonthlyDataLoad  
Function Equals ProcessExecuteEx  
Action Equals Kill
```

IBM Cognos Performance Management Hub

Deprecated in v2.0.9.3 IBM Cognos Performance Management Hub (PMHub) is a common server framework under which IBM Cognos performance management products interoperate. PMHub provides common application services to IBM Cognos Analysis for Microsoft Excel, Operations Console, and other TM1 client applications.

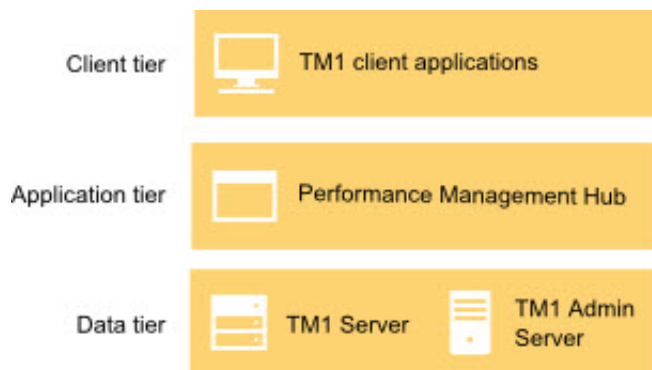


Figure 5. IBM Cognos Performance Management Hub

PMHub provides a web-based user interface, which you can use to set configuration options.

Opening the Performance Management Hub web page

Deprecated in v2.0.9.3 You can set configuration options by using the IBM Cognos Performance Management Hub web page.

Procedure

1. In a web browser, go to the following URL: `http://[server_name]:9510/pmhub/pm/admin`

For example, if you are currently logged into the computer where the TM1 Application Server is running, go to `http://localhost:9510/pmhub/pm/admin`.

The Administration and Configuration page is displayed.

2. Expand **configurations** and select a category.
The configuration settings are displayed.
3. To change a value, double-click a cell in the Value column, type a value, and then press Enter.

Configuring logging

Deprecated in v2.0.9.3 You can adjust the default message logging settings for IBM Cognos Performance Management Hub. You can also enable additional logs.

Performance Management Hub (PMHub) logs messages through the WebSphere® Liberty Profile, according to the setting defined in the `server.xml` file. The `server.xml` file is located in `<installation location>\wlp\usr\servers\tm1\`.

By default, all ERROR and WARNING log messages are written to `./logs/tm1_messages.log`. The `tm1_messages.log` file contains all messages, except trace messages, that are written or captured by the logging component. When the file size reaches 20 MB, the file rolls over to `./logs/tm1_messages_{datestamp}.log`. Only two files are kept.

The following attributes of the `<logging>` element define the default log settings in the `server.xml` file:

```
consoleLogLevel="WARNING"
logDirectory="{wlp.user.dir}/../logs"
messageFileName="tm1_messages.log"
maxFiles="2"
maxFileSize="20"
```

You can adjust these settings, for example, to change the number of log files that are kept, change the value of `maxFiles`.

You can also enable additional logging.

console.log

Contains redirected standard output and standard error messages from the JVM process. The default log level is AUDIT.

trace.log

Contains all messages that are written or captured by the product.

For example, to produce advanced trace logging at the finest level, for all components that start with `com.ibm.ba`, add the following attributes to the logging element in the `server.xml` file:

```
traceFileName="pmhub_trace.log"
traceFormat="ADVANCED"
traceSpecification="*=audit:com.ibm.ba.*=finest"
```

The attributes of the element are then as follows:

```
consoleLogLevel="WARNING"
logDirectory="{wlp.user.dir}/../logs"
messageFileName="tm1_messages.log"
maxFiles="2"
maxFileSize="20"
traceFileName="pmhub_trace.log"
traceFormat="ADVANCED"
traceSpecification="*=audit:com.ibm.ba.*=finest"
```

For more information about WebSphere logging, see [Logging and Trace](#) on IBM Knowledge Center.

Using the TM1 Top utility

The TM1 Top utility dynamically monitors the threads running in an instance of the TM1 server. TM1 Top can also display information about the sandboxes and job queue for a specific server.

This utility is similar to the UNIX top utility, which allows dynamic monitoring of the processes running on a given system.

TM1 Top is a stand-alone utility that runs within a console (command) window on a Microsoft Windows system. It is designed to make minimal demands on the TM1 server and the supporting network and system.

Except for a user-initiated login process, TM1 Top does not use any cube or dimension resources in the TM1 server, and does not use or interact with the data or locks on the TM1 server. The server-side processing that supports TM1 Top runs in a separate light thread to allow TM1 Top to report server state even if the server is unresponsive to users.

Note: The TM1 Top user interface and help text are not translated and display in English in non-English versions of TM1.

Displaying user names in TM1 Top

The **MaskUserNameInServerTools** parameter determines if user names are displayed or masked out in the TM1 Top console display.

When `MaskUserNameInServerTools=TRUE` is set in the `Tm1s.cfg`, user names are masked in the TM1 Top for security purposes. To unmask the names, the administrator can use the **V** (verify) command to log in as an administrator.

This parameter is set to `FALSE` by default. If you do not explicitly set this parameter to `TRUE`, the TM1 Top console displays user names even when administrator access has not been verified.

Configuring the Tm1top.ini File

Before running TM1 Top, edit the initialization file `Tm1top.ini`.

The `Tm1top.ini` file is an ASCII file that specifies environment information for the TM1 Top utility.

By default, a sample `Tm1top.ini` file is installed to the `install_dir\bin` directory. When you run TM1 Top, the `Tm1top.ini` file should be located in the same directory as the TM1 Top executable file.

A sample of a configured `Tm1top.ini` file is shown below.

```
adminhost=
servername=planning sample
logfile=c:\temp\tm1top.log
logperiod=50
logappend=T
refresh=10
adminsvrsslcertid=
adminsvrsslcertauthority=
adminsvrsslcertrevlist=
exportadminsvrsslcert=
adminsvrssllexportkeyid=
```

If you configure TM1 to use SSL for secure data transmission, then you must configure the SSL parameters in the `Tm1top.ini` file. For details about configuring TM1 Top to work with SSL, see [“Configuring the Tm1top.ini File”](#) on page 82.

Do not include any spaces between the parameter name and the parameter value when editing the `Tm1top.ini` file.

The parameters in the `Tm1top.ini` file are described in the following table.

Parameter	Description
AdminHost	Machine name or IP address of the computer running the Admin Server. Default value is localhost.

Parameter	Description
LogFile	<p>Specifies the path and file to which the log information is written. As TM1 Top is running, the status lines can be continuously written to this ASCII file so that the activity over time can be monitored.</p> <p>Format is: <code>logfile=path-to-log-file</code></p> <p>where <i>path-to-log-file</i> must specify the complete path, filename, and file extension. A default filename and file extension is not provided.</p>
LogPeriod	<p>Specifies the time interval between updates being written to the log file. This value should be a multiple of the Refresh time parameter.</p> <p>The format for this parameter is: <code>logperiod=nn</code></p> <p>where <i>nn</i> is the number of elapsed seconds between updates to the log file. Setting this to zero disables logging.</p> <p>For example, if the screen Refresh is set to 2 seconds, LogPeriod could be set to 10 seconds so that every fifth screen display will be output to decrease the amount of data written to the file.</p> <p>The default value is 2 seconds.</p>
LogAppend	<p>Specifies if log data is appended to the log file that is set with the LogFile parameter, or if the file is overwritten when a new session is started. If the LogAppend parameter is not set, the existing log file will be overwritten.</p> <p>Valid values are T and F as shown in the following examples:</p> <p>If you set <code>LogAppend=T</code>, log data is appended to the log file specified by the LogFile parameter.</p> <p>If you set <code>LogAppend=F</code>, the existing log file will be over-written.</p> <p>The default value is F, which overwrites the existing log file.</p>
Refresh	<p>Time interval between refreshing of the TM1 Top display. Format is:</p> <p><code>refresh=nn</code></p> <p>where <i>nn</i> is the time interval in seconds.</p> <p>Default value is 2 seconds.</p>
ServerName	<p>TM1 server name. This is the ServerName from the server configuration file, Tm1s.cfg.</p>

Note: You can also enter these values at the command prompt when starting TM1 Top, overriding the values in the Tm1top.ini file. For details, see [“Running TM1 Top with command line options”](#) on page 84.

Running TM1 Top

To run TM1 Top, edit the Tm1top.ini file to set the parameters you want to use then run the .exe command.

Procedure

1. Edit the initialization file, Tm1top.ini, as described in [“Configuring the Tm1top.ini File”](#) on page 82.
2. Place the Tm1top.exe and Tm1top.ini files in the `install_dir\bin` directory.

Note: By default, the Tm1top.exe file and a sample Tm1top.ini file are installed in the `install_dir\bin` directory when you install the TM1 server.

3. Run the Tm1top.exe file from a command prompt to start TM1 Top.

```
tm1top.exe
```

By default, the Tm1top.ini file is read from the same directory that contains the TM1 Top executable. To specify a different location for the Tm1top.ini file, use the -i command-line parameter as follows.

```
tm1top.exe -i path-to-initialization-file
```

Running TM1 Top with command line options

You can over-ride the basic parameter values in the Tm1top.ini file by entering them on the command line when starting TM1 Top.

Note: SSL security parameters can not be entered at the command line and must be specified in the Tm1top.ini file if TM1 is configured to use SSL. For details, see "Configuring the TM1 Top Utility to Use SSL."

Use the following format to run TM1 Top with command-line options:

```
tm1top.exe -OptionName1 OptionValue1 -OptionName2  
OptionValue2 ..
```

Where OptionName and OptionValue can be any of the following parameter and value combinations:

- -adminhost *admin-host-name*
- -servername *server-host-name*
- -refresh *refresh-period*
- -logfile *file-path*
- -logperiod *nnn*
- -logappend T or F

For example, to run TM1 Top with the ServerName parameter set to sdata and the refresh parameter set to 5 seconds, enter the following:

```
tm1top.exe -servername sdata -refresh 5
```

Note: Use quotes for parameter values that include spaces, as follows:

```
tm1top.exe -servername "planning sample"
```

To show a list of available parameters, use the /? option as follows:

```
tm1top.exe /?
```

For more details on TM1 Top parameters, see [“Configuring the Tm1top.ini File” on page 82](#).

Viewing and understanding the TM1 Top display

When TM1 Top is running, it displays a set of fields and status information.

Each row in the display represents one unique thread running in the TM1 server that you are monitoring. The title bar of the console window displays the current values for the AdminHost, ServerName, and Refresh parameters.

Note: To see more lines or a wider display, you can re-size the console window or use a smaller font size. If the display fills the entire height of the console window, you can use the up and down arrow keys on your keyboard to go to the next or previous page within the console window.

TM1 Top can run in the following different display modes by pressing the related key when the program is running:

- Normal (**n**)

- Sandbox (**s**)
- Queue (**q**)

Normal display mode

In normal display mode, status fields are displayed by TM1 Top.

Field Name	Description
ID	The thread ID of the underlying system thread in the TM1 server.
User Name	<p>The name of the user or process that started the thread. This can be one of three possible thread types.</p> <ul style="list-style-type: none"> • User Thread - Name of an actual user that is logged into TM1. A value of ??? means the user is currently in the login stage. <p>Note: The MaskUserNameInServerTools parameter determines if user names are displayed or masked out in the TM1 Top console display. For details, see “Displaying user names in TM1 Top” on page 82</p> <ul style="list-style-type: none"> • Chore Thread - A chore running on the TM1 server. The name of the chore thread is displayed in the following format: <i>C:chore-name</i>
User Name (continued)	<p>The name of the process running under the chore is displayed in the Function field.</p> <ul style="list-style-type: none"> • System Thread - A TM1 system process running on the TM1 server. Displayed in the following format: <i>Th:name of system thread</i> • Th:Pseudo - The Pseudo thread is used to clean up user-defined consolidation (UDC) objects. • Th:Stats - The Stats thread represents the thread for the performance monitor feature that is started when a user selects the Start Performance Monitor option in TM1 Architect and Server Explorer. • Th:DynamicConf - This thread dynamically reads and updates parameters in the TM1 server configuration file, tm1s.cfg.
State	<p>The current processing state of a specific thread. This value can be one of the following values:</p> <ul style="list-style-type: none"> • Idle • Run • Commit • Rollback • Wait • Login <p>For more information, see “Understanding TM1 thread processing states” on page 87.</p>
Function	<p>The current API function that the thread is executing.</p> <p>If the thread is running a chore, as opposed to a function for an interactive user, the Function field will display the name of the process that is currently executing under that chore.</p>

Field Name	Description
Obj Lock Status	<p>Lock status of the last object that was locked.</p> <p>Displayed in the format: <i>R# IX# W#</i></p> <p>Where:</p> <ul style="list-style-type: none"> • <i>R</i>, <i>IX</i>, and <i>W</i> represent the different lock modes on the object. For more information, see “Understanding TM1 lock modes” on page 87. • <i>#</i> is the number of locks for each specific lock mode. <p>Note: : The name of the object is displayed in the State field.</p>
User Lock Status	<p>Total lock status and count for all objects being used by a thread.</p> <p>Displayed in the format: <i>R# IX# W#</i></p> <p>Where:</p> <ul style="list-style-type: none"> • <i>R</i>, <i>IX</i>, and <i>W</i> represent the different lock modes on the objects. For more information, see “Understanding TM1 lock modes” on page 87. • <i>#</i> is the total number of locks, across all objects, for each specific lock mode.
Time	<p>The total time, in seconds, that the current API function or chore process has been processing.</p> <p>The name of the API function or chore process is displayed in the Function field.</p>

Sandbox display mode

You can switch TM1 Top to sandbox mode.

Pressing the **s** key while TM1 Top is running switches to sandbox mode. Pressing **n** returns you to the normal TM1 Top display.

In addition to the ID and Username fields, the following fields become available in sandbox mode:

Fields	Description
Active Sandbox	Name of the currently active sandbox.
Total Sandbox Memory	The Total Sandbox Memory being consumed for all sandboxes by user.
Number of Sandboxes	Number of sandboxes for this user.

Queue display mode

You can switch TM1 Top to queue mode.

Pressing the **q** key while TM1 Top is running switches to queue mode. Pressing **n** returns you to the normal TM1 Top display.

In addition to the ID and Username fields, the following fields become available in queue mode:

Fields	Description
Name	Name of the sandbox in the queue.
Node Name	Name of the node for the sandbox in the queue.

Fields	Description
Status	Date, time, and state when the status was taken for the sandbox in the queue.
Time in Queue	Length of time sandbox has been in the queue before it was processed.

Understanding TM1 lock modes

TM1 uses a set of three lock modes to control access to TM1 data.

When the TM1 server is running, these locks are applied to individual objects, such as cubes, views, and dimensions, as these objects are accessed.

TM1 Top displays the status of these locks for the objects being used by the current threads running in a TM1 server. Lock status is displayed by TM1 Top under the State, Obj Lock Status, and Total Lock Status fields.

The lock modes for TM1 objects are described in the following table.

Lock Mode	Description
R	R-lock (read lock) - A shared lock that allows many threads to read from an object at the same time, but does not allow another thread to modify or write to this object until all R-locks have been released.
IX	IX-lock (intent-to-write-lock) - A lock that reserves the right for a thread to obtain a W-lock on an object when all R-locks have been released. Only one thread at a time is allowed to have an IX-lock on an object.
W	W-lock (write lock) - An exclusive lock that allows only one thread at a time to access and write changes to an object. No other thread can read or modify this object until the W-lock has been released.

Understanding TM1 thread processing states

TM1 Top displays the current processing state of each thread in the State field.

A thread can be in one of the following processing states.

State	Description
Idle	Thread is currently not processing.
Run	Indicates the thread is actively running. Also displays the lock mode and object being accessed by the thread in the following format: Run: <i>Lock Mode-(Object Name)-Object Type</i> Where: <ul style="list-style-type: none"> • Lock Mode is one of the possible object lock modes; R, IX, or W. For more information, see “Understanding TM1 lock modes” on page 87. • Object Name is the name of the last TM1 object that was being acted on by the API function displayed in the Function field. • Object Type is a numeric value representing the type of TM1 object.
Commit	Thread is currently committing changes to the objects it was accessing.

State	Description
Rollback	Indicates the thread encountered a conflict while committing and could not proceed. The thread will automatically re-try until the conflict is resolved.
Wait	<p>Thread is currently waiting for a specific condition to complete. Displayed in the format:</p> <p>Wait: <i>Wait Status</i> - (<i>Object Name</i>) - <i>Object Type</i></p> <p>Where:</p> <ul style="list-style-type: none"> • Wait Status can be one of the following conditions: <ul style="list-style-type: none"> – WR - (WaitForWriterEvent) Thread is waiting for R-locks to be released so it can obtain a W-lock on the object. – IXR - (WaitForIXReaderEvent) Thread is waiting for a W-lock to finish so it can get either an R-lock or an IX-lock on an object. – IXC - (WaitForIXConflictEvent) Thread is requesting an IX-lock, but is waiting for another thread with an IX-lock on the same object to finish and release the lock. – IXCur - (WaitForIXCurrentEvent) Thread is requesting an IX-lock for an object, but is waiting for a thread with a R-lock on the same object to release its lock. – WC - (WaitForCompletionEvent) Thread is waiting for another thread to complete and release its locks. – DRR - (Data Reservation Release) Thread is waiting for a Data Reservation to be released. • <i>Object Name</i> is the name of the last TM1 object that was being acted on by the API function displayed in the Function field. • <i>Object Type</i> is a numeric value representing the type of TM1 object.
Login	A user is currently logging into the TM1 server.

TM1 Top commands

TM1 Top provides a command prompt to interact with the utility.

You can enter the following commands at the TM1 Top command prompt identified by Cmd : _ in the top left corner:

Command	Description
N	Displays TM1 Top normal (thread status) mode.
S	Displays TM1 Top sandbox mode.
Q	Displays TM1 Top queue mode.
W	Writes the current TM1 Top information to a file.
X	Exits the TM1 Top utility.
H	Displays help text.

Command	Description
V	Verifies a TM1 Top administrator session. Displays a user name and password prompt so the TM1 administrator can log into the current session of TM1 Top and enter commands to cancel processing in other threads.
C	Cancels processing in another thread. Note: Before using the Cancel command, an administrator must first log into the current session of TM1 Top by using the V (Verify) command.
K	Cancels a sandbox submission in the job queue for a specific job ID number. Requires administrator login with the V (Verify) command.
Up, Down arrow keys	Use the up and down arrow keys to scroll the status display if there are more lines in the status than the height of the console window.

Canceling a thread's processing

A system administrator can use TM1 Top to request the cancellation of certain thread operations or functions that are processing-intensive.

TM1 Top can cancel only threads with a state of **Run** that are running functions involving TurboIntegrator processing, constructing a view, or calculating a view.

The state of a thread is indicated under the **State** field in the TM1 Top display.

This functionality provides a way to cancel an action that is negatively impacting other threads by consuming too much of the server's processing time. For example, a TI process that gets stuck in a loop, or when a user tries to compute a very large and demanding view.

To cancel a thread, you must use the **Cancel** and **Verify** commands together.

Understanding the Cancel process

The Cancel process cancels a running thread.

When a thread is in a Run state, and the process involves TurboIntegrator processing, constructing a view, or calculating a view, then the thread can be canceled. While running, the thread will check to see if its "exit indicator" has been set, and if set, the thread will exit out of the current operation.

Understanding the Verify process

As a compromise between a full login procedure and no security at all, TM1 Top requires that the TM1 Top user must first be "verified" as having administrator access before the Cancel command can be used.

TM1 Top was designed to run without requiring a user to log on to the server. This allows TM1 Top to run against a server which might be resource deadlocked. If a login procedure was required, then the login process would also become deadlocked because it requires access to cubes and other data that are manipulated under the resource locks.

However, it is not desirable to allow just any user running TM1 Top to cancel user threads. This verification process is initiated using the V (verify) command. The process can be done in advance and is maintained for the life of the TM1 Top connection.

Using the Verify and Cancel commands

You can verify a TM1 Top session and cancel a thread.

Procedure

1. Enter V to display the following prompt to verify a TM1 Top session:

```
Verify Admin Access
```

```
Admin User Name:
```

2. Enter the login name of an admin user.

The following prompt displays:

```
Verify Admin Access
```

```
Admin Password:
```

3. Enter the password for the admin user.

If you entered a valid admin user name and password, then the following confirmation displays:

```
Admin access verified
```

```
Press any key to continue:
```

4. After the session has been verified, enter C to initiate the Cancel process.

The following prompt displays:

```
Cancel processing of a thread
```

```
Thread-id:
```

5. Enter the ID of the thread to be canceled. You can find the thread ID displayed in the first column of the TM1 Top display.

Other debugging tools

You can use either ADPlus or Dr. Watson as a debugging tool to collect server crash information on a Windows system. You can also collect files or set up client logging.

Installing and running ADPlus

ADPlus is part of the Microsoft Debugging Tools for Windows package.

You can download the most recent package at <http://www.microsoft.com/whdc/devtools/debugging/default.msp>.

Both a 32-bit version and a 64-bit version of Debugging Tools for Windows is available, be sure to download the version that is appropriate for your operating system.

For complete details on ADPlus, including system requirements, see <http://support.microsoft.com/kb/286350>.

Running ADPlus directly on the TM1 Server

Run ADPlus directly on the computer where the TM1 server is installed.

Procedure

1. Start the TM1 server.
2. Open a command prompt window and navigate to the Debugging Tools for Windows directory.
3. Open the Windows Task Manager and note the process ID (PID) for Tm1s.exe (if running the server as an application) or Tm1sd.exe (if running the server as a service).
4. At the command prompt, enter `adplus -crash -p [PID]`

For example, `adplus -crash -p 492`

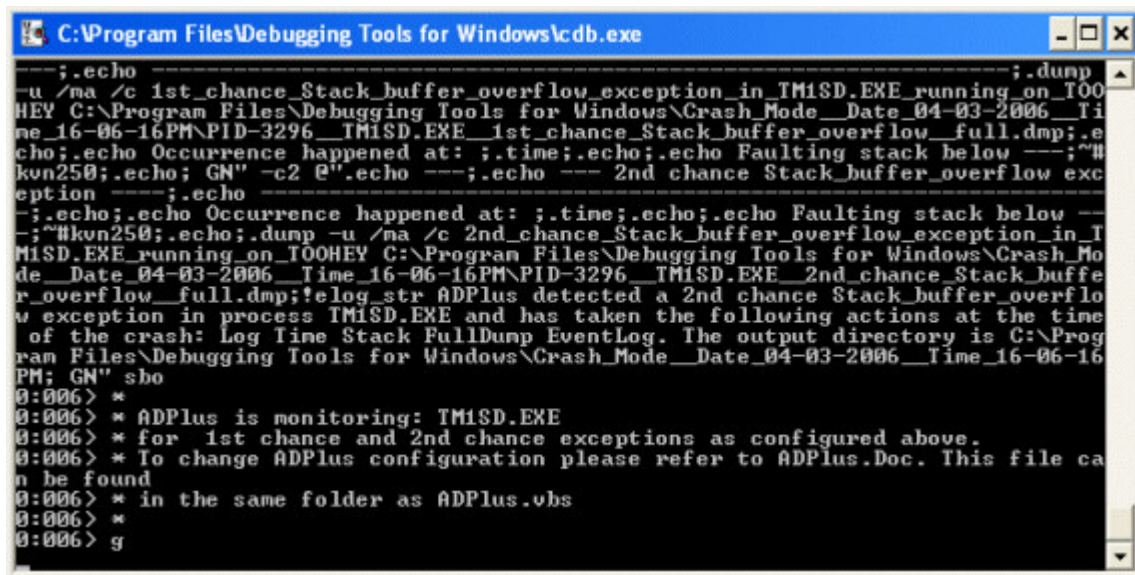
If you cannot determine the process ID for the server you want to monitor, you can also use one of the following commands to start ADPlus:

- `adplus -crash -pn tm1sd.exe` if running the TM1 server as a service
 - `adplus -crash -pn tm1s.exe` if running the TM1 server as an application
5. A warning message appears indicating that an environment variable is not set. You do not have to take any action based on this warning. Click **OK** to dismiss the warning.

Another message appears indicating that a new subdirectory will be created in the Debugging Tools for Windows directory. The new subdirectory, which is named `Crash_Mode_DateStamp_TimeStamp`, receives the dump files that are generated when the TM1 server crashes.

6. Click **OK** to dismiss the message.

Note that a new command prompt window is now open on your desktop. The Microsoft Console Debugger (`cdb.exe`) runs in this window, as shown in the following image.



```
C:\Program Files\Debugging Tools for Windows>cdb.exe
;.echo -----;.dump
-u /na /c 1st_chance_stack_buffer_overflow_exception_in_TM1SD.EXE_running_on_TOOHEY C:\Program Files\Debugging Tools for Windows\Crash_Mode_Date_04-03-2006_Time_16-06-16PM\PID-3296_TM1SD.EXE_1st_chance_stack_buffer_overflow_full.dmp;.echo Occurrence happened at: ;.time;.echo;.echo Faulting stack below ---;~#kvn250;.echo; GN" -c2 e".echo ---;.echo --- 2nd chance Stack_buffer_overflow_exception -----;.echo
-;.echo;.echo Occurrence happened at: ;.time;.echo;.echo Faulting stack below ---;~#kvn250;.echo;.dump -u /na /c 2nd_chance_stack_buffer_overflow_exception_in_TM1SD.EXE_running_on_TOOHEY C:\Program Files\Debugging Tools for Windows\Crash_Mode_Date_04-03-2006_Time_16-06-16PM\PID-3296_TM1SD.EXE_2nd_chance_stack_buffer_overflow_full.dmp;!elog_str ADPlus detected a 2nd chance Stack_buffer_overflow_exception in process TM1SD.EXE and has taken the following actions at the time of the crash: Log Time Stack FullDump EventLog. The output directory is C:\Program Files\Debugging Tools for Windows\Crash_Mode_Date_04-03-2006_Time_16-06-16PM; GN" sbo
0:006> *
0:006> * ADPlus is monitoring: TM1SD.EXE
0:006> * for 1st chance and 2nd chance exceptions as configured above.
0:006> * To change ADPlus configuration please refer to ADPlus.Doc. This file can be found
0:006> * in the same folder as ADPlus.vbs
0:006> *
0:006> *
0:006> g
```

Capturing .dmp files from the TM1 Server

When the server crashes, three dump files (.dmp) are created in the `Crash_Mode_DateStamp_TimeStamp` subdirectory. IBM needs the `...2nd_chance_AccessViolation...` file to debug the server crash.

Using CTRL-C to force a TM1 Server crash

You can force a server crash. Enter **CTRL-C** in the `cdb.exe` command prompt window.

The name of the resulting dump (.dmp) file will include the string `...1st_chance_CONTROL_C_OR_Debug_Break...`. This is the file IBM needs to debug the TM1 server.

Taking a snapshot of current server state

You can take a snapshot of the current server state, while leaving the TM1 server running:

Procedure

1. Start the TM1 server.
2. Open the Windows Task Manager and note the process ID (PID) for `Tm1s.exe` (if running the TM1 server as an application) or `Tm1sd.exe` (if running the TM1 server as a service).
3. Open a command prompt window and `cd` to the Debugging Tools for Windows directory
4. At the command prompt, enter `adplus -hang -p <PID>`.

For example, `adplus -hang -p 492`

Note: You can run a `-hang` command even when `adplus` is already running in `-crash` mode.

A warning message appears indicating that an environment variable is not set. You do not have to take any action based on this warning.

5. Click **OK** to dismiss the warning.

Another message appears indicating that ADplus is running in HANG mode. The message also indicates that a subdirectory will be created in the Debugging Tools for Windows directory. The new subdirectory, which is named `Hang_Mode_DateStamp_TimeStamp`, receives the dump files that record the current server state.

6. Click **OK** to dismiss the message.

A new dump file is immediately generated, recording the current server state.

7. Open the new subdirectory in the Debugging Tools for Windows directory and note the presence of a single dump (.dmp) file. This is the file IBM needs to debug your server.

Running ADPlus remotely from a TM1 client

You can run ADPlus on a different computer. In some environments, you might want to manage and run ADPlus from the computer on which a TM1 client is running, rather than directly from the TM1 server.

Installing and running ADPlus on the server

You can install and run ADPlus on the computer where the TM1 server is running.

Procedure

1. Install the Microsoft Debugging Tools for Windows package.
2. Create a batch file named `Remoteshell.cmd` in the debugging tools installation directory. This batch file should contain the following single line:

```
C:\Program Files\Debugging Tools for Windows (x86)\remote.exe /s "cmd.exe"
remoteshell
```

3. From a command prompt, enter the following line:

```
at 2 minutes past current time, 24 hour format C:\Program Files\Debugging Tools for Windows
(x86)\remoteshell.cmd
```

For example, if your system clock shows 4:45 PM, enter the following line at the command prompt:

```
at 16:47 C:\Program Files\Debugging Tools for Windows (x86)\remoteshell.cmd
```

This command will run the `remoteshell.cmd` batch file at 4:47 PM

4. After the specified time, run the `at` command from a command prompt. Do not include any parameters to the command, type only `at`.

If the `remoteshell.cmd` batch file ran successfully, you should see a report that lists the Status, Day, Time and Command Line listing such as "c:\Program Files\Debugging Tools for Windows \remoteshell.cmd."

For more details on the `at` command, see <http://technet2.microsoft.com/windowsserver/en/library/40b9beb1-3578-48f9-93e1-7ca6760c1c151033.mspx>.

5. Open the Windows Task Manager and note the process ID (PID) for `Tm1s.exe` (if running the TM1 server as an application) or `Tm1sd.exe` (if running the TM1 server as a service). You will need to know the PID when running ADPlus from the TM1 client.

Running ADPlus from a TM1 client

You can run ADPlus from a computer on which a TM1 client is installed.

Procedure

1. Copy the `remote.exe` and `breakin.exe` files from the debugging tools installation directory on the TM1 server computer.
2. Save `remote.exe` and `breakin.exe` to an easily identified directory on the computer running the TM1 client. These instructions assume that `remote.exe` is saved in `C:\debuggers`.

3. Open a command prompt and cd to the C:\debuggers folder.
4. Enter the following command:

```
remote.exe /c remote server name remoteshell
```

where *remote server name* is the name of the computer on which the TM1 server is installed; it is not the name of the TM1 server. For example, if smithers is the name of the computer where the server is installed, you would enter the command

```
remote.exe /c smithers remoteshell
```

5. You are now running commands on the server computer from the command prompt on the client computer. To confirm that you are properly connected to the server, enter `dir C:\` at the command prompt. You should see the contents of the C drive on the server.
6. From the command prompt, cd to the debugging tools installation directory on the TM1 server computer.
7. Enter the following at the command prompt:

```
Adplus -crash -quiet -p <PID>
```

PID is the process ID of the TM1 server. For example, if the process ID for the server is 1588, enter `Adplus -crash -quiet -p 1588` at the command prompt.

You should receive confirmation that the debugger is attached to the TM1 server.

ADPlus is now ready to collect debugging information and will continue to monitor the server until it crashes.

Stopping ADPlus

You can stop ADPlus. Usually, you will want to let ADPlus run until a server crash occurs. However, if you need to stop ADPlus, enter `breakin.exe <PID>` at the command prompt on the client computer.

For example:

```
breakin.exe <PID>
```

PID is the process ID of the TM1 server. For example, if the process ID for the TM1 server is 1588, enter `breakin.exe 1588` at the command prompt to stop the debugging process.

Running Dr. Watson

Dr. Watson is part of the Windows operating system, so you do not need to install the tool.

Procedure

1. From the Windows taskbar, click **Start**, then click **Run**
2. In the Run dialog box, enter `d\wtstn32`, then click **OK**.
3. In the Dr. Watson dialog box, the Log File Path can be set to any path and Crash Dump can be set to any path/filename you want. Take note of the path/filename you set for the Crash Dump, as this file must be sent to IBM for analysis.
4. Set and enable only the following options:
 - Number of Instructions: 10
 - Number of Errors to Save: 10
 - Crash Dump Type: Full
 - Dump Symbol Table
 - Dump All Thread Contents
 - Create Crash Dump File
5. Click **OK** to begin running Dr. Watson.

Setting Dr. Watson as the default debugging tool

If you have ADPlus or another debugging tool installed on your system, you may need to set Dr. Watson as the default debugger.

Procedure

1. From the Windows taskbar, click **Start**, then click **Run**.
2. In the Run dialog box, enter `drwtsn32 -i`, then click **OK**.

You can now confirm that Dr. Watson is running as the default debugger.

3. From the Windows taskbar, click **Start**, then click **Run**.
4. In the Run dialog box, enter `regedit`, then click **OK**.
5. From the Registry Editor, click **Edit**, then click **Find**.
6. Search for **AeDebug** in the registry.

The settings for the AeDebug key should show `drwatsn32` set as your Debugger.

tm1xfer

The `tm1xfer` utility compresses and moves TM1 server objects from one platform to another platform while preserving mixed case names for objects on both Microsoft Windows and UNIX platforms.

tm1xfer syntax

The `tm1xfer` tool uses one parameter.

```
tm1xfer <directory> | <file> | -v | -h | -?
```

Only the first parameter is processed. Any extra parameters are ignored by the application. The meaning of each parameter is explained in the following table. Any incorrect parameter or invalid file name and path generates an appropriate error message.

Parameter	Description
<directory>	Specifies the path and name of the TM1 database directory that contains the files to be modified and compressed for transfer.
<file>	Specifies the path and name of the compressed file which contains all the database files. This file is decompressed into a new directory with the same name.
-v	Displays the version number of the <code>tm1xfer</code> tool.
-h	Displays comprehensive help documentation for the application.
-?	Displays a short help dialog, which includes usage syntax and version number.
no parameters	Displays the short help dialog (same as -?).

Compressing directories with tm1xfer

To compress the TM1 database files, invoke the `tm1xfer` tool at the command line followed by the path and name of the directory that contains the TM1 database files. The `tm1xfer` tool recognizes that it is a directory and compresses it.

A new file is created with the extension ".zip" that contains all the compressed database files. The name of the new file is the name of the directory specified at the command line.

The name of the original directory is also stored inside the .zip file. If the .zip file already exists, the `tm1xfer` tool alerts you that a file with the same name already exists and prompts you to indicate whether the new file should overwrite the older file. If you choose yes, then the `tm1xfer` tool proceeds to compress the database files and overwrite the older .zip file. Otherwise, the application will exit without changing any files.

The tm1xfer tool performs the following actions:

- tm1xfer does the necessary platform specific handling of file names and file objects to make sure that every file is converted to this format during the compression process. All subdirectories in the main database directory are recursively compressed.
- tm1xfer ignores raw store files and they are not included in the zipped file. If the raw store files are detected, the tm1xfer alerts you to process audit log events manually before migrating.
- tm1xfer ignores dimension difference files and they are not included in the zipped file.
- tm1xfer stores the files inside the zipped file differently than a common zip application. Therefore, when decompressing TM1 files that were previously compressed with the tm1xfer tool, using a common zip application such as WinZip, the resulting decompressed files result in unusable files. This is done to discourage the use of other zip applications, since the tm1xfer tool does extra file processing that is not done by other compression utilities.

Compression Example

The following example assumes that there is a TM1 database called Sales Planning on a TM1 Server on a Microsoft Windows platform, and that the database directory on disc is called sales_planning.

```
C:\Users\obaluch\Documents\dbfile>dir

Volume in drive C has no label.
Volume Serial Number 4C9A-1CE1

Directory of C:\Users\obaluch\Documents\dbfile

05/11/2011  05:54 PM    <DIR>          .
05/11/2011  05:54 PM    <DIR>          ..
05/11/2011  06:11 PM    <DIR>          sales_planning
             0 Files(s)          0 bytes(s)
             3 Dir(s)      91,081,957,376 bytes free
```

To move the database files to a AIX® server, invoke the following command at the command prompt on the Windows platform:

```
C:\Users\obaluch>tm1_xfer sales_planning
```

This action results in the following files:

```
C:\Users\obaluch\Documents\dbfile>dir

Volume in drive C has no label.
Volume Serial Number 4C9A-1CE1

Directory of C:\Users\obaluch\Documents\dbfile

05/11/2011  05:54 PM    <DIR>          .
05/11/2011  05:54 PM    <DIR>          ..
05/11/2011  06:11 PM    <DIR>          sales_planning
05/11/2011  06:24 PM    <FILE>          43,611,174 sales_planning.zip
             1 Files(s)          43,611,174 bytes(s)
             3 Dir(s)      91,038,346,202 bytes free
```

The tm1xfer tool recognizes that sales_planning is a directory with database files, and compresses it into a ".zip" file. The result is another file called sales_planning.zip.

Decompressing directories with tm1xfer

To decompress a zipped file containing TM1 database files, invoke the tm1xfer tool at the command line followed by the path and name of the zipped file. The tm1xfer tool recognizes that it is a zipped file and decompresses it.

A directory is created with the same name as the original directory which was compressed, which is usually the same as the .zip file, but not always. This is true because the name of the original directory is stored inside the .zip file, and the output directory is named using the stored name. If the name of the zip filename is changed to something different, it still uses the original name.

If a directory with the same name already exists, the tm1xfer tool issues a warning message and prompts you to indicate if it should be overwritten. If you select *yes*, the application deletes the current directory and creates an empty directory with the zip file name. If you select *no*, the application exits without changing or decompressing any files.

Inside the main database directory, the tm1xfer tool reconstructs the original subdirectory hierarchy.

The database files are decompressed inside the directory and processed to handle mixed case object names and Unicode based on the platform and locale. On Windows platforms, the files are stored with mixed case file names encoded using UTF-16. The files do not require embedded object names. Since the files inside the .zip file are stored using mixed case and using UTF-16, there is usually no extra conversion necessary.

File paths, the }applications file, and the workbooks in the folder of the same name must be edited so that any references with paths containing "/" must be converted to paths containing "\".

On UNIX platforms, the files are stored using an 8-bit encoding such as UTF-8 or ISO-8859-1, and all lowercase. If UTF-8 is used, then Unicode characters are encoded using that system. Otherwise, Unicode characters are converted to the form U+xxxx, where xxxx is the UTF-16 code point for that character.

The tm1xfer tool does not automatically modify the tm1s.cfg file. You must manually change any settings such as database path. The tm1xfer tool does not automatically modify the server name, for example the server name inside Microsoft Excel workbook applications.

Decompression Example

On an AIX system, the zipped file called sales_planning.zip is copied over to location on the filesystem where the database directory should be located. To extract the database files, run the following command on AIX:

Issue command

```
bash-3.2$ ls
aix64          tests          sales_planning.zip
```

```
bash-3.2$ tm1xfer sales_planning.zip
```

After:

```
bash-3.2$ ls
aix64          tests          sales_planning          sales_planning.zip
```

The tm1xfer tool recognizes that sales_planning.zip is a zipped file, and proceeds to unzip the database files and to modify the filenames and files to handle mixed case objects and Unicode encoding. This is done based on the current platform and locale.

Notes on tm1xfer

Keep in mind the following other considerations when using tm1xfer.

The tm1xfer tool compresses and decompresses most files located in the database folder, except the following types: raw stores and dimension differences.

Important: If raw stores are present, run the "Process Audit Log Events..." command for that TM1 server.

Cases are not preserved in pre-Unicode files

When pre-Unicode files exist in the database, the tm1xfer log displays **Pre-Unicode file format detected in <filename>**. To fix this situation, resave the view, subset, and dimension on in Microsoft Windows operating system then run the tm1xfer tool again. Then transfer the files to UNIX and run the tm1xfer tool again to unzip the database.

Archives created using another application than the tm1xfer tool

The tm1xfer tool can be used to pack and move database files between two TM1 servers running on the same type of platform, for example from Windows to Windows. In this case, the files are compressed in the tm1xfer format, and then decompressed for Windows and the current locale. It is important to remember that tm1xfer filters out a few file types. These include raw stores and dimension differences, so these will be lost.

tm1xfer archives files in a format different from other compression utilities such as WinZip or WinRar. For this reason, archives created using tm1xfer cannot be decompressed using other compression utilities, and vice versa.

If a complete copy of the database files has to be moved between two systems on the same platform type *and* the same locale, it may be a better idea to use a compression utility that doesn't do any file processing, such as WinZip or tar. On the other hand, the tm1xfer tool should be used when moving database files between systems running on the same platform but different locales, since it will modify the files based on locales; for example, between two UNIX systems, where one is using en_US.utf8 locale and another en_US.iso88591

Platforms with older version of TM1 server

The tm1xfer tool validates if the file version is at least version TM1 version 9.1 and above. If it is a file version lower than 9.1, it ignores the file.

Handling of private applications

Although the content of private Applications is transferred, and any references to TM1 object files within the application (for example, within a websheet) are modified to ensure the reference remains valid on UNIX, the tm1xfer tool does not retain the original names of private applications that are migrated to UNIX from Windows.

Server name in tm1s.cfg

If the name of the TM1 server is different after a database is transferred to another host machine – and it normally is – the server name must be manually changed in the tm1s.cfg file. tm1xfer does not make this change automatically.

Server names in Workbooks

Custom applications often contain many workbooks that contain references to the TM1 server name. This name must change when a TM1 database is transferred between heterogeneous platforms (Windows to UNIX or UNIX to Windows) and will often change when moving between servers on the same platform. tm1xfer does not provide any assistance in changing server names within workbooks.

Error messages for tm1xfer

The following table lists the error message you may encounter when using tm1xfer.

Message	Description	Remedy
This locale is not supported by your Java™ installation	The filesystem encoding is not supported by this program.	Update your Java installation so it supports the filesystem encoding
<arg> does not exist	The file or directory name provided in the command line argument is a file or directory that does not exist.	Ensure that the file or directory name on the command line is correct.
Cannot resolve <file>: <message>	There was a problem resolving the full path to the file.	Use the specific information in the message to determine the source of the problem.

Table 4. Error Messages for tm1xfer (continued)

Message	Description	Remedy
Cannot resolve <base_dir>: <message>	There was a problem resolving the full path to the base directory.	Use the specific information in the message to determine the source of the problem.
Cannot serialize out list of files: <message>	There was an error writing this Java object to disk.	Use the specific information in the message to determine the source of the problem.
Cannot open <filename> for reading: <message>	tm1xfer does not have permission to read this file in the database directory.	Check your file system permissions. Use the specific information in the message to determine the source of the problem.
Error writing to zip file: <message>	There was a problem writing to the zip file.	Make sure the disk is not full and that tm1xfer has appropriate write permissions. Use the specific information in the message to determine the source of the problem.
Error closing zip file: <message>	There was a problem closing the zip file. Usually this problem resolves itself when tm1xfer exits.	Use the specific information in the message to determine the source of the problem.
Error opening zip file: <message>	tm1xfer checks for a missing file; however, Java requires a FileNotFoundException log message.	Use the specific information in the message to determine the source of the problem.
<filename>: Not a valid archive	The passed argument exists but it is not a valid archive that was created by tm1xfer	Ensure the file name on the command line is correct.
Error reading zip file: <message>	There was an error reading the zip file. The file may have been corrupted or there may be a permission problem.	Make sure the zip file transferred correctly and tm1xfer has read access to it. Use the specific information in the message to determine the source of the problem.
Error reading <filename>	There was an unspecified error while reading the current file	Use the specific information in the message to determine the source of the problem.
Error interpreting path: <message>	There was a problem resolving this path	Use the specific information in the message to determine the source of the problem.
File read error: <message>	There was a problem reading this file.	Ensure that tm1xfer has read permission for the file.

Table 4. Error Messages for tm1xfer (continued)

Message	Description	Remedy
Error reading Application Entries dimension	There was an error reading }ApplicationEntries.dim in the database.	If you cannot open }ApplicationEntries.dim using TM1, then the file could be corrupt. If the file is not corrupt, then ensure the file's permissions give tm1xfer read access.

Usage notes for tm1xfer

These notes may be helpful when using tm1xfer for the first time.

- On Microsoft Windows, the command is executed in CMD.EXE (or compatible command processor). On UNIX, use any suitable shell (for example, C-shell or Bash).
- The standard TM1 install does not add the TM1 server bin directory to the users PATH variable. Since you normally will run this tool in the directory containing the TM1 data files (which are not the bin directory), you may encounter the error "'tm1xfer' is not recognized as an internal or external command, operable program or batch file."

There are three ways to handle this situation:

1. Fully qualify the command name with the tm1 server bin path name; for example: "c:\Program Files\IBM\cognos\tm1\bin\tm1xfer"
2. Temporarily add the tm1 server path to the PATH variable using the SET command; for example set PATH=%PATH%;c:\Program Files\IBM\cognos\tm1\bin\tm1xfer
3. To permanently set the TM1 server path to the PATH variable: right-click **My Computer**, > **Properties**, **Advanced** tab, click **Environment Variables**, scroll to find PATH in the System Variables, click **Edit**, then append the TM1 server bin path to the existing path using a semicolon delimiter; for example: ;c:\Program Files\IBM\cognos\tm1\bin\tm1xfer.

The steps may be slightly different in your operating system.

- **Note:** When specifying Windows path or filename in CMD.EXE, the string may have to be enclosed in double quotes if it contains any spaces in the name. For example, to execute tm1xfer on a folder called Data Files, you need to specify tm1xfer "Data Files" not tm1xfer Data Files.
- tm1xfer takes a folder name as a starting point and packs everything in that folder and below it. Typically, it would be used on the Data directory only. Co-resident with the Data folder might be other files, such as tm1s.cfg and log files. You may have put these in special folders. If these files need to be copied to the target platform, then a temporary folder should be created somewhere that can be used to contain a copy of the Data folder and all the other files to be transferred. tm1xfer can then be used to pack up that temporary folder.
- Two typical use case scenarios are: (a) moving a TM1 database (and application data) to a different server and (b) deploying a TM1 database developed on one platform (such as Windows) to different platform (such as Linux® or AIX). In the case of (a) you probably want to move everything .log files, tm1s.cfg etc. In the case of (b), you probably just want to copy the data folder. In the case of (a) you might need to make changes to tm1s.cfg (for example, if drive letters or paths are different on the target machine. In the case of (b), change tm1s.cfg because the path conventions are different (backslash versus slash etc.)
- tm1xfer generates messages. These can be directed to a file by using the CMD.EXE or UNIX Shell redirection option; for example, tm1xfer data > tm1xfer.log
- The TM1 server should be shut down before running tm1xfer to pack up the files. tm1xfer sometimes detects that the TM1 server was not properly shut down, and will prompt the user to determine whether they should continue or not.

Collecting files for analysis following a TM1 Server crash

If the TM1 server crashes, you must collect the following files for IBM support and engineering to analyze:

- The dump file generated by your debugging tool.

If you are running ADPlus, the dump file is named

ProcessID_ProcessName_2nd_chance_AccessViolation_TimeStamp.dmp. For example, PID-492_TM1SD.EXE_2nd_chance_AccessViolation_full_ODB8_2008-03-08_10-17-59-052_01EC.dmp

If you are running Dr. Watson, the dump file name and location was set with the Crash Dump option when you started Dr. Watson.

- The Tm1server.log file. This file is located in the TM1 server data directory.
- All TM1ProcessError.log files generated during the server session that crashed.

When a process error log file is generated, TM1 assigns a unique name that lets you readily identify which TurboIntegrator process generated the error file and the time at which the file was created. File names are assigned using the convention *TM1ProcessError_time stamp_process name.log*. In this convention, *time stamp* is the time (expressed as *yyyymmddhhmmss* GMT) at which the file was generated and *process name* is the name of the TurboIntegrator process that caused the errors.

There may be multiple TM1ProcessError.log files associated with the server session that crashed. All TM1ProcessError.log files are stored in the server data directory.

- Do not collect the TM1 transaction log (Tm1s.log) unless specifically requested by IBM support.

Client logging

A client-side logging tool is available that can collect data about activity for TM1 clients, such as IBM Cognos TM1 Architect and IBM Cognos TM1 Perspectives.

Due to the large amounts of data that can be collected, this tool should only be used under the direction of customer support.

Here is a sample client log:

```
log4j.rootLogger=DEBUG, R1
#Server Explore
log4j.logger.COriionMainApp=DEBUG
log4j.logger.CCubeView=DEBUG
log4j.logger.CubeViewGrid=DEBUG
log4j.logger.COriionTreeView=DEBUG
log4j.logger.COriionApp=DEBUG
log4j.logger.CMultiSelect=DEBUG
log4j.logger.CCreateSandbox=DEBUG
log4j.logger.loggerViewDimension=DEBUG
#Subset Editor
log4j.logger.CSubsetWnd=DEBUG
log4j.logger.CSubsetEntry=DEBUG
log4j.logger.CSubsetListProp=DEBUG
log4j.logger.CSubsetListBox2=DEBUG
log4j.logger.CSubsetListView1=DEBUG
log4j.logger.CSubsetRegExp=DEBUG
log4j.logger.CSubsetSelByAttr=DEBUG
log4j.logger.CSingleSelect=DEBUG
#ISB
log4j.logger.CXLCtrlsApp=DEBUG
log4j.logger.CConnectServerDlg=DEBUG
log4j.logger.CViewCtrl=DEBUG
log4j.logger.CGetViewDlg=DEBUG
log4j.logger.CubeViewerISB=DEBUG
log4j.logger.ExcelReportPrintDPAGE=DEBUG
log4j.logger.CTM1Connector=DEBUG
log4j.logger.ExcelReportPrintIncludedSheetsPage=DEBUG
log4j.logger.CAutoExcel=DEBUG
log4j.logger.SubsetEditorIntLog=DEBUG
#Perspectives
log4j.logger.TM1Perspectives=DEBUG
log4j.logger.TM1AF=DEBUG
log4j.logger.TM1ExcelAPI=DEBUG
log4j.appender.R1=org.apache.log4j.RollingFileAppender
log4j.appender.R1.File=c:\ClientDebugLog.log
```

```
log4j.appender.R1.MaxFileSize=10 MB
log4j.appender.R1.MaxBackupIndex=50
log4j.appender.R1.layout=org.apache.log4j.PatternLayout
log4j.appender.R1.layout.ConversionPattern=%X{pid}
%t %p %d{%-m-%d %H:%M:%S,%Q} %c %m%n
log4j.appender.R1.layout.TimeZone=GMT
```

To turn on client-side logging, create a file called `tm1p-log.properties` in the `C:\Documents and Settings\Administrator\Application Data\Applix\TM` location on the client machine. The `TM1p.ini` file must be in the same folder.

Be sure that location is also set in the `log4j.appender.R1.File=c:\\ClientDebugLog.log` line of the log file.

If you put the log files directly on the C drive as in the example, `log4j.appender.R1.File=c:\ClientLog.log`

Be sure to use `\\` slashes.

If you want to use a temp directory, specify the location using the other slash:

```
log4j.appender.R1.File=c:/temp/ClientLog.log
```

Be sure to update the file parameter with the new location if you change the `tm1p-log.properties` file, for example:

```
location.log4j.appender.R1.File=c:\<new location>\ClientDebugLog.log
```

If the presence of this file and at least one component within it is set to `DEBUG`, client-side logging is collected. `log4j.rootLogger=DEBUG` can only be set to `OFF` or `DEBUG` (which turns client logging ON).

To turn off logging, remove or rename this file. You can turn off logging for an individual component by changing the value from `DEBUG` to `OFF`.

To customize the log file, you can change:

- Location of the log file is set in `log4j.appender.R1.File=c:\\ClientDebugLog.log`
- Log file max size is set in `log4j.appender.R1.MaxFileSize=10 MB`. When this size is met a new log file is created.
- Backup index is set in `log4j.appender.R1.MaxBackupIndex=50`.

odbc_test tool

The `odbc_test` tool is used to diagnose and test an IBM Cognos TM1 ODBC connection on UNIX.

Use `odbc_test` to test an ODBC access code in exactly the same way that the TM1 server uses this code. One of the process switches is the path to the TM1 server configuration file directory, similar to the TM1 server's `-z` switch so that exactly the same ODBC library can be loaded on UNIX using `odbc_test`.

Syntax

```
odbc_test -z <path to config directory>
```

```
[-dbname <db name> -user <username> -passwd <passwd> -query <select query> -n  
#-of-rows-to-retrieve]
```

```
[-n number-or-records]
```

```
[-t (do timing)]
```

```
[-r number-repetitions]
```

```
[-h] prints the help file.
```

Use `-u` to make the underlying SQL calls use the wide Unicode flavor of the ODBC interface.

Specify a `-t` switch to time the calls running this with just the path to the configuration directory to just load the library and list the available data sources.

This program will run the specified query, and lists the first 10 records. The file `odbc_test.ini` will be read before the command line switches are processed. The fields in this file are:

`config_dir:`

`path-to-config-directory:`

`dataset_name:`

`user_name:`

`user_passwd:`

`sql_query:`

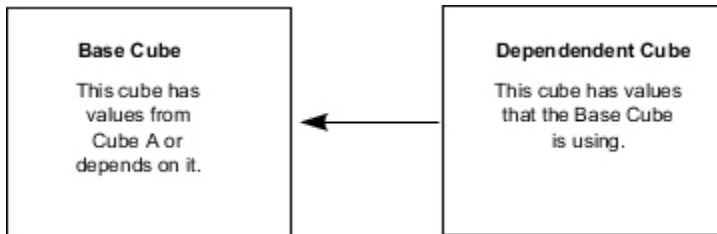
The `sql_query:` line and all subsequent lines are concatenated to make the SQL query.

Chapter 8. Understanding cube dependency

IBM TM1 server establishes dependencies so it can properly invalidate cube caches.

Dependencies are established generally when a user executes a query on a cube and when that query includes a rule-calculated value from another cube. A dependency can be established to one or many cubes based on the scope of the rule within the Base cube.

Here is a simple example of a dependency within TM1:



In this simple case, the caches in Base cube are invalidated when:

- there is a change in the Base cube itself
- or
- there is a change in the dependent cube

Since a dependency is established, when there is a change in the dependent cube, the Base cube must also change.

Without this dependency map, the Base cube data could become inconsistent with the Dependent cube. The Base data is dependent upon the data in the dependent cube so any change in the Dependent cube requires the Base cache to change also.

Changes in dependency mapping

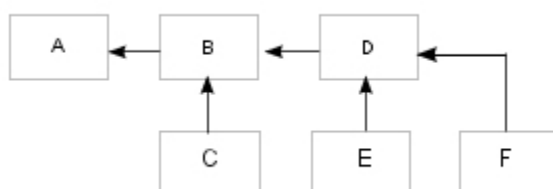
This topic describes cube dependency mapping.

Dependency behavior now follows these rules:

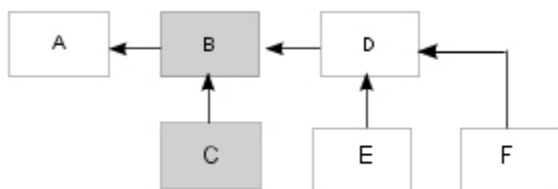
- Dependencies once established are NOT cleared based on data invalidation.
- Cube dependencies can be established ahead of time using the AddCubeDependency function.

When dependencies are created, the dependency lists are protected by an object lock between the cubes affected.

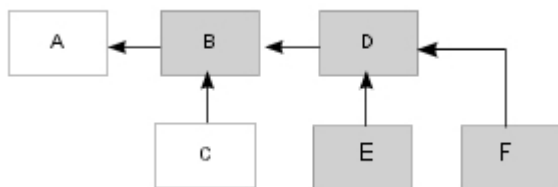
The fact that dependency lists are protected by an object lock and "hangs" off a cube has an important effect on application upgrade. The dependency blocks writers as well as readers from a cube object when a reader queries a cube and when there is a need to add or write to the dependency list object for that cube. This lock persists for the duration of the read query. For fast queries, this impact is minimal and a one time "cost" for each cube-to-cube dependency. For slower queries, however, other users may be blocked for usually long periods. In complex applications, the dependency object may be locked many times initially as users execute only certain rules. Here is an example:



In this example, the arrows show data flow based on rule connections. Cube B has a two rules: one that pulls from Cube C and the other pulls from Cube D. Cube D has a single rule that pulls from Cubes E and F.



When User 1 reads from Cube B, a rule calculated there requires data pulled from Cube C. So Cube B, C and related cube objects are blocked until User 1's query is completed.



When User 2 reads from Cube B, a rule calculated there requires data pulled from Cube D. Cube B, D, and related cube objects are now blocked until User 2's query is completed.

To address this situation, the TurboIntegrator function AddCubeDependency was created to establish dependencies. This function allows users to establish dependencies after events such as a re-start of the server.

Clearing cube dependencies

The following table describes the TM1 events or actions that destroy a cube's dependency.

Event	Dependency Action
Server Shutdown	Cube dependency lists are destroyed.
Rule Compile (Note: any dimension update causes a rule compile.	Eliminates only those cubes where rules have been deleted or modified.

When cube dependencies are cleared, depending on the application and the level of user activity, you may want to establish those cube dependencies manually.

Creating cube dependencies

The following table describes the TM1 events or actions that add a cube dependency.

Event	Dependency Action
Server Startup	Calculated Feeders will create dependencies.
Server Startup + Persistent Feeders	Persistent Feeders also store dependency maps for each cube.
Rule Compile	Calculated Feeders for the particular cube rule will create dependencies.
Rule Calculation	Create dependencies based which rule is executed by the query.
TI Function	TI Function can create dependencies between cubes.

Using AddCubeDependency to establish dependencies

The AddCubeDependency function is used to create a manual dependency between two cubes.

AddCubeDependency is a TurboIntegrator function that is valid only in TurboIntegrator processes.

This function creates a manually-created dependency between Cube A and Cube B.

Syntax: AddCubeDependency('CubeA', 'CubeB');

Argument	Description
CubeA	The name of the base cube.
CubeB	The name of the dependent cube.

Cube B relies on a rule that is dependent on Cube A.

This function can be set in anywhere in the TurboIntegrator Advanced tab.

Debugging and identifying cube dependencies

Using TurboIntegrator to preset dependencies may be necessary.

To identify existing cube dependencies requires an in-depth understanding of an application so that you can analyze all of the rules for each cube including control cubes. Even when you have that level of knowledge about an application, it is important to turn on the following logging capability to identify cube dependencies as they are established.

For example, when the following command is added to the the tm1s-log.properties file:

```
log4j.logger.TM1.Cube.Dependency=DEBUG
```

the following information is returned that identifies dependencies:

```
INFO 2010-06-09 18:09:23.728 TM1.Cube.Dependency
    Adding cube dependency: cube
    "{ElementSecurity_}Cubes" depends on cube "{CubeSecurity}"
```

The TurboIntegrator command for this would be:

```
AddCubeDependency('{CubeSecurity}', '{ElementSecurity_}Cubes');
```

Chapter 9. Replicating cubes

You can replicate cubes from one IBM TM1 server to another, and synchronize the updates across the copied cubes. Depending on your access privileges, you can copy cubes (and their associated dimensions, rules, subsets, and views) from one server to another, and synchronize the updates among the copied cubes either at specified time intervals or on demand. The process of copying cubes from one server to another is called replication.

Note: Replication and synchronization operations in IBM Planning Analytics should be performed only by members of the ADMIN group. Members of other predefined groups (DataAdmin, SecurityAdmin, OperationsAdmin) do not have all the required access privileges to perform these operations.

Advantages of Using Replication

Replication offers the following advantages.

- Enhances response time because users can update a cube locally without having to communicate across a network.
- Lets users access and update a copy of a cube, even when they are not connected to the remote server on which the original cube resides.
- Greatly enhances the scalability of TM1.

TM1 provides bidirectional synchronization for replicated cubes. During the synchronization process, TM1 copies the data updates and metadata from the original cube to its replicated versions, and copies the data updates from the replicated versions back to the original cube.

Considerations when using replication

The following considerations apply to replication:

TM1 versions

All TM1 Servers in a replication process must be the identical version.

Remote servers

You can replicate cubes that reside on remote servers only. You cannot replicate cubes that reside on local servers.

Local servers

TM1 clients can replicate cubes to their local server only if they are running that server as an independent process. The machine must have a network card. To run a local server as an independent process, clients need to select the Local Server Execution Mode: Independent Process option in the TM1 Options dialog box.

Access privileges

When you replicate a source cube on a remote server to a local server, any elements to which the local client has NONE access on the remote server, will have a value of zero. If the client has READ (or higher) access to a consolidation that includes elements to which the client has NONE access, the consolidation will appear to be the sum of only those elements to which the client has READ (or higher) access. The consolidation, as reported to the client, will not be the sum of all elements, as in the source cube.

Note: When you set security levels and establish replication to the local servers, you must know the implications of replicating data to which a client does not have at least READ access.

Tm1s.cfg file

The Tm1s.cfg file must be configured to register the target and source servers with the same TM1 Admin Server. For more information, see [“Configure the tm1s.cfg file to support replication” on page 111](#).

Length of directory path and cube name

The total length of the path name for the target TM1 Server's data directory and the name of the cube you are replicating cannot exceed the Windows path name limit of approximately 256 characters. If this limit is exceeded, due to a long path name or cube name, TM1 displays the following error message: Could not register the cube.

Transaction Logging

If you are performing a synchronization process, transaction logging must be enabled for the mirror cubes on the target server that are part of the replication and synchronization process. If you are performing a bidirectional synchronization, transaction logging must be enabled for all the related cubes on both the source and target servers.

TM1 uses these log files to track the changes made to the source and mirror cubes. To verify that transaction logging is enabled, see [“Enabling and disabling transaction logging” on page 45](#).

CubeProperties Control Cube

The values stored in the CubeProperties control cube are specific to a TM1 Server and are not copied from the master to the target server during a replication process. For example, if you want the Measures dimension for a replicated cube to be set on the target server, you must manually set the value in the CubeProperties control cube on the target server.

Relationships created by replication

A replication creates a relationship between two cubes and between two servers. There are two types of relationships: cube and server. A replication creates server configurations.

Cube relationships

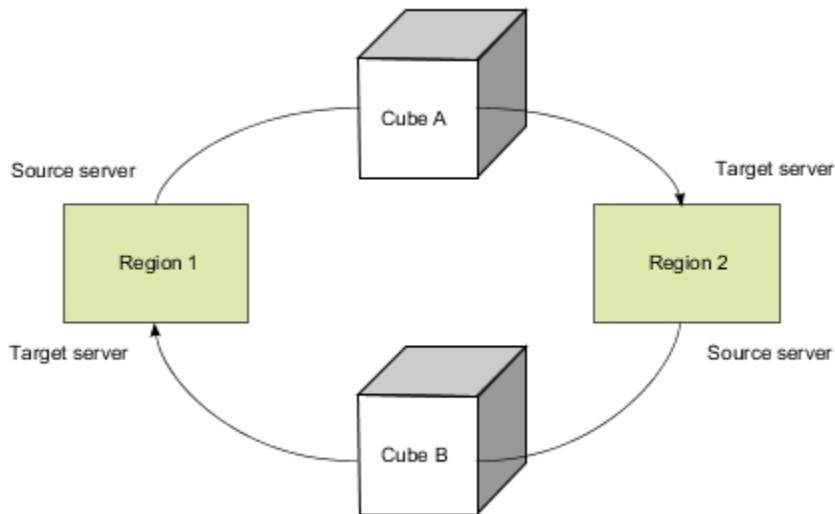
When you replicate a cube, the original cube is called the source cube, and a copy of that cube is called the mirror cube. You can replicate a single cube on many different servers, and you can replicate a replicated cube.

Server relationships

To replicate a cube, you must log on to a remote server and create a replication connection. This connection establishes the remote server that you just logged on to as a source server and the server you logged in from as the target server.

After you establish a replication connection, you can replicate as many cubes as you want to through this connection. TM1 uses the logon ID for the replication connection to determine your access rights to the source data.

The same TM1 server can be both a target server for some cubes, and a source server for other cubes. In this situation, there would be two replication connections between the target and source servers, as illustrated in the following diagram.



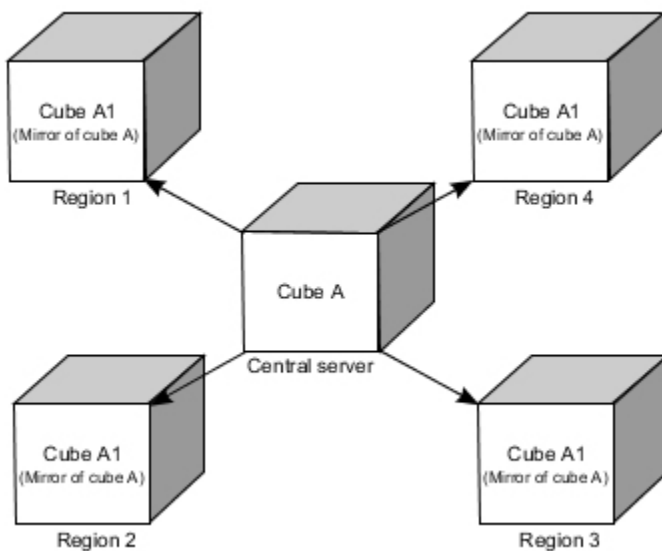
The synchronization process occurs at the server level, rather than at the cube level. When you synchronize the replication connection, TM1 updates all the mirror cubes that are part of the same replication connection. For example, if you replicate Cubes A and B from a central server to the Region 4 server, both of these cubes are updated when you initiate the synchronization process for the replication connection between these two servers.

Typical server configurations

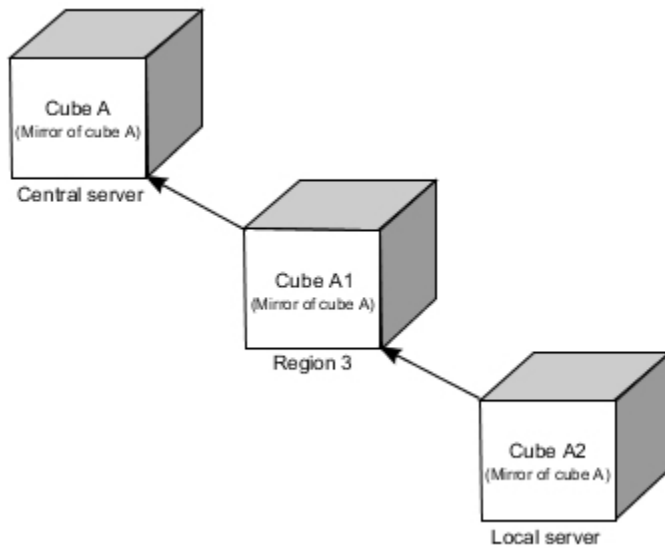
The typical server configurations that TM1 create when you replicate cubes are shown in the following diagram.

Note: As the TM1 administrator, you must ensure that no loops are created through a replication.

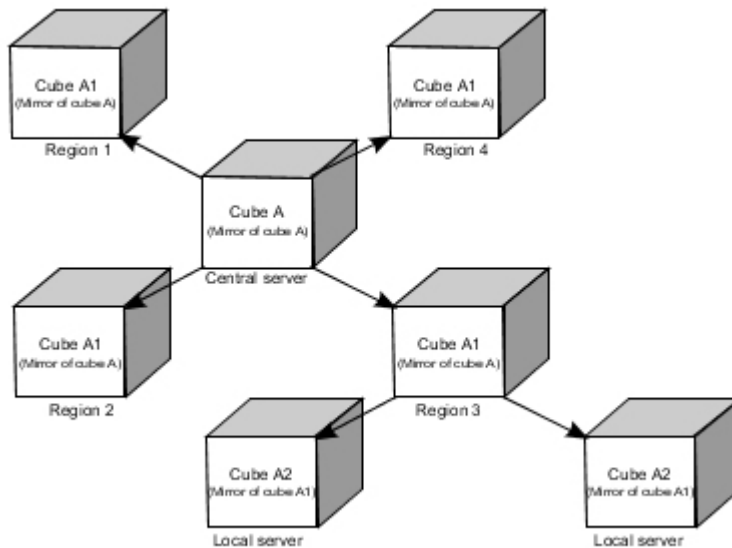
When you replicate a single cube on many different servers, you create a star configuration of servers. For example, you can replicate Cube A from a central server to four regional servers.



When you replicate a replicated cube, you create a chain configuration of servers. For example, using the star configuration of servers, Cube A can be replicated from Region 3 to a local server.



You can combine the star and chain configurations, as shown below.



Required access privileges

To replicate a cube, you need the following access privileges.

- Read access or higher to the cube you want to replicate.

If you have access privileges for only certain elements in a cube, TM1 copies the values for those elements, but does not populate the rest of the cube.

- Admin access to the server where you are creating the replicated cube. You must be in the ADMIN group on that server. Members of the predefined groups (DataAdmin, SecurityAdmin, OperationsAdmin) do not have all the required access privileges to perform replication and synchronization operations.

Users have Admin access to their local servers and can replicate any cube for which they have Read access to the local server.

As the TM1 administrator, you are responsible for replicating cubes on the remote servers.

Note: Although you can replicate a cube if you have Read access to it, you need Reserve access to the cube to write updates back to cube during the synchronization process. See [“The synchronization process”](#) on page 114.

Set up replication

There are three stages you must follow to set up a replication for a TM1 system.

Procedure

1. Create a replication connection between the target and source servers. For more information, see [“The replication process”](#) on page 112.
2. Replicate one or more cubes from the target server to the source server.
3. After you make updates to the cubes, initiate the synchronization process, if synchronization is done on demand. For more information, see [“The synchronization process”](#) on page 114.

Configure the tm1s.cfg file to support replication

When you establish a replication connection, the target and source TM1 servers must be registered with the same TM1 Admin Server. Each server can be registered with other Admin servers, but the target and source servers must share at least one common Admin Server. If the two servers do not use the same Admin Server, the replication and synchronization results are unpredictable.

Procedure

1. Edit the AdminHost parameter in the Tm1s.cfg file for the target TM1 server so it registers with the same Admin Server as the source.

```
AdminHost=<hostname1>;<hostname2>
```

where:

- <hostname1> is the computer name or IP address of the TM1 Admin Host where the target TM1 server is registered.
 - <hostname2> is the computer name or IP address of the TM1 Admin Host where the source TM1 server is registered.
2. You can use a mix of computer names and IP addresses and you can also list multiple Admin hosts:

For example:

```
AdminHost=boston;newyork
```

or

```
AdminHost=192.168.1.17;192.168.1.22
```

or

```
AdminHost=boston;192.168.1.17;192.168.1.22;myserver;192.168.1.40
```

What to do next

For more information, see also [Chapter 2, “TM1 Admin Server operations,”](#) on page 9.

The replication process

After you create a replication connection between two servers, you can replicate the cubes from the source server to the target server.

TM1 allows the replication of control cubes, including these cubes:

```
}DimensionProperties
```

```
}CubeProperties
```

```
}ConnectionProperties
```

In most circumstances, you should not replicate these cubes. If you do, be aware that the cubes on the target server overwrite the cubes on the source server when synchronization occurs, thereby removing all the replication and synchronization settings that were made through the UI.

Replicate a cube

When a cube is replicated, the total length of the path name for the target TM1 server's data directory and cube name can not exceed the Microsoft Windows path name limit of approximately 256 characters. If this limit is exceeded, due to a long path name or cube name, TM1 displays the following error message: Could not register the cube.

Procedure

1. Double-click the **Replications** icon for the target server.
The tree expands to list existing replication connections.
2. Double-click the replication connection you want to use.
The tree expands to list the cubes that you can replicate.
3. Right-click the cube you want to replicate and click **Replicate**.
The Replicate Cube dialog box opens.
4. Fill in the Replicate Cube dialog box as described in [“Specifying information about the replicated cube” on page 112](#).
5. Click **OK**.

TM1 replicates the cube, as described in [“What happens when TM1 replicates a cube” on page 114](#).

Note: After you click **OK** to replicate a cube, you cannot change the specifications for this particular replication. However, you can create another replicated version of the source cube with different specifications.

Specifying information about the replicated cube

The **Replicate Cube** dialog box lets you specify the following information about the mirror cube you are creating:

- Name of the mirror cube.
- Name of each dimension in the mirror cube.
- Whether the mirror cube uses dimensions that are replicated from the source server or local dimensions.
- Whether the dimensions replicated from the source server are synchronized when the cubes are synchronized.
- Whether the mirror cube uses rules that are replicated from the source cube.
- Whether the rules replicated from the source server will be synchronized when the cubes are synchronized.

The Replicated Cube dialog box that displays when you replicate the depletion cube from the inventory server is shown in the following figure. The replication status might be different, depending on whether you have already replicated the dimensions in the cube.

Note: The sections that follow describe how to change the TM1 default settings. We recommend that you do not change any default settings unless you fully understand the implications of doing so.

Cube information

By default, TM1 names the mirror cube with the source cube name.

Procedure

1. To specify another name for the mirror cube, enter the name in the Name field.

Note: Do not rename a cube if you are replicating rules in that cube.

2. To replicate the cube and allow later synchronization, select **Copy Data and Set to Synchronize**.
3. To replicate the cube but disable later synchronization, select **Copy Data but Do Not Set to Synchronize**.

Dimension information

By default, TM1 specifies that each dimension in the mirror cube will be copied from the source server, and then synchronized when the mirror cube is synchronized with its source. TM1 does not change the dimension names when it copies them. You can override the default settings for a dimension when you replicate a cube.

Procedure

1. To rename a dimension, slowly click the dimension twice (do not double-click), and type the new name.



CAUTION: Do not rename a dimension if you are replicating rules in the associated cube.

2. To direct TM1 to copy a dimension but not synchronize it, select the dimension, and clear the Set Dimension to Synchronize option.
3. To direct TM1 to use a local dimension instead of a replicated dimension:
 - Double-click the dimension to open the Select Dimension for use in Replicated Cube dialog box.
 - Select the dimension you want to use and click **OK**.



CAUTION: By default, TM1 overwrites the local dimension with the source dimension. If you do not want TM1 to overwrite the dimension, select the Don't Overwrite Dimension option. This automatically directs TM1 to treat the local dimension as a stand-alone dimension.

Rule information

By default, TM1 specifies that each rule in the mirror cube is copied from the source cube, and then synchronized when the mirror cube is synchronized with its source.

Procedure

1. To specify that you want TM1 to copy the rules but not synchronize them, clear the Set Rule to Synchronize option.
2. To specify that you do not want TM1 to copy rules for the mirror cube, select the **Do Not Copy Rule** option.

View information

By default, TM1 specifies that each public view in the mirror cube is copied from the source cube, and then synchronized when the mirror cube is synchronized with its source.

Procedure

To specify that you do not want TM1 to copy views for the mirror cube, clear the **Replicate Views** option.

Subset information

By default, TM1 specifies that each public subset in the mirror cube is copied from the source cube, and then synchronized when the mirror cube is synchronized with its source.

Procedure

To specify that you do not want TM1 to copy subsets for the mirror cube, clear the **Replicate Subsets** option.

What happens when TM1 replicates a cube

When TM1 replicates a cube, the following action occurs.

- Optionally, TM1 replicates the cube dimensions. When you initiate the replication process, you decide which dimensions you want to replicate, and of those, which dimensions you want to synchronize when you synchronize the cubes. You can also choose to use one or more local dimensions with a replicated cube.
- Optionally, TM1 replicates the cube rules. When you initiate the replication process, you decide which rules you want to replicate, and of those, which rules you want to synchronize when you synchronize the cubes.
- Represents the cube as an icon under the Cubes icon on the target server.
- Names the source server in the Replication Server field on the attributes bar for each replicated cube.
- Names the source server in the Replication Server field on the attributes bar for each replicated dimension.

The synchronization process

You always initiate the synchronization process from the target server, which is the server that contains the mirror cubes. You must have administrator authority for this server to initiate the synchronization process.

You can synchronize on demand or schedule the synchronization process to be automatic.

- TM1 updates the data in all the mirror cubes that are part of the same replication connection. You cannot specify individual cubes to update.
- TM1 also updates the replicated dimensions if you made changes to their definitions and rules.
- TM1 always writes updates from the source cubes to the mirror cubes.
- TM1 does the following with updates you make to the mirror cube:
 - Writes the updates back to the source cube, if the updates were made by users with Reserve access to the source cube.
 - Does not write the updates back to the source cube, if the updates were made by users with Read or Write access to the source cube.
- TM1 uses the transaction log files to keep track of the changes made to the source and mirror cubes. During the synchronization process, TM1 uses the log files to determine which updates to make.
 - If you are performing a synchronization process, transaction logging must be enabled for the mirror cubes on the target server that are part of the replication and synchronization process.
 - If you are performing a bi-directional synchronization, transaction logging must be enabled for all the related cubes on both the source and target servers.

To verify that transaction logging is enabled, see [“Enabling and disabling transaction logging”](#) on page 45 in the IBM Cognos *TM1 Operations* documentation.

If there is an update conflict for a particular cell, TM1 uses the most recent update, no matter where this update was made (that is, in the source cube or the mirror cube).

When a synchronization occurs, either as the result of a scheduled chore or manual execution, a log file that records all the actions that occur during the synchronization is created in the logfiles directory on the target TM1 server. The synchronization log file is named TM1ReplicationLog_<time stamp>_<source server>.log. For example, TM1ReplicationLog_20140308035140_SData.log.

The synchronization log records the time at which each action within the synchronization occurs, the name of the user who established the synchronization, and a brief description of the action.

```
20140308041525 [Admin] => Synchronization to "Planning Sample" started.
20140308041525 [Admin] => Current R&S values: LastSyncTime=20140308041233
LastSyncTimeMaster=20140308041233 LastSyncStarRecord=20140308040749.
20140308041525 [Admin] => Start: Replication of objects not yet copied.
20140308041525 [Admin] => End: Replication of objects not yet copied.
20140308041526 [Admin] => Replicating dimension "plan_version".
20140308041526 [Admin] => Synchronizing cube: (P)"plan_BudgetPlan" => (S)"plan_BudgetPlan"
20140308041526 [Admin] => WARNING: No changes found in Planet server.
20140308041526 [Admin] => Synchronization of cube "plan_BudgetPlan" completed: Star <= 0
20140308041526 [Admin] => Synchronizing cube: (P)"promotions" => (S)"promotions"
20140308041526 [Admin] => Synchronization of cube "promotions" completed: Star <= 12
20140308041528 [Admin] => Synchronizing cube: (S)"plan_BudgetPlan" => (P)"plan_BudgetPlan"
20140308041528 [Admin] => WARNING: No changes found in Star server.
20140308041528 [Admin] => Synchronization of cube "plan_BudgetPlan" completed: Planet <= 0
20140308041528 [Admin] => Synchronizing cube: (S)"promotions" => (P)"promotions"
20140308041528 [Admin] => Synchronization of cube "promotions" completed: Planet <= 30
20140308041530 [Admin] => Synchronization to "Planning Sample" completed.
LastSyncTime=20140308041527 LastSyncTimeMaster=20140308041526
LastSyncStarRecord=20140308041234.
```

Synchronize on demand

To synchronize updates on demand, complete the following steps.


Procedure

1. In the Server Explorer, access the server that contains the mirror cubes.
2. Double-click the Replications icon.
3. Right-click the icon for the replication connection for whose cubes you want to synchronize, and click **Synchronize All Data**.

Schedule synchronization

You can create a chore to automatically synchronize data at a regular interval using TurboIntegrator.

Procedure

1. In the Server Explorer, select the **Chores** icon beneath the target server for which you want to automate synchronization.
2. Right-click the **Chores** icon and click **Create New Chore**.
The Chore Setup Wizard opens.
3. Select the replication for which you want to automate synchronization.
4. Click the arrow button  to move the replication from the Available list to the Selected List.
5. Click **Next**.
The second screen of the **Chore Setup Wizard** displays.
6. Use the calendar and Time field to set a start date and time for the initial synchronization.
7. Use the fields in the Chore Execution Frequency box to set the interval at which the synchronization should be executed.
8. Click **Finish** to complete the scheduling.

Synchronize over unstable or wide area network connections

You can use the `MaximumSynchAttempts` and `SyncUnitSize` parameters to improve the stability of a synchronization process that is running over an unstable network connection such as a long distance wide area network (WAN) with high latency, poor bandwidth and poor transmission quality.

For more details, see the `MaximumSynchAttempts` and `SyncUnitSize` parameters in *Planning Analytics Installation and Configuration*.

Maintaining replication connections


To replicate cubes, you must first create a replication connection between two servers.

Create a replication connection

To create a replication connection between two servers, complete the following steps.

Procedure

1. Open the Server Explorer.
2. Make sure that the target server is visible.
3. Double-click the Replications icon for the target server.

For example, to replicate a cube from the `sdata` server to the `financiareporting1` server, double-click the Replications icon  under the `financiareporting1` server.


The Create Server Replication Object dialog box opens.


Note: If the server already has a replication connection, you must select the Replications icon, right-click the icon, and click **Insert New Replication** to open the Create Server Replication dialog box.

4. Fill in the dialog box as follows:
 - Select the source server in the **From Server** box.
 - Specify the username and password with which you want to connect to the source server.
 - If the source server uses Cognos 8 security (CAM authentication), enter the Cognos 8 Namespace ID of the source server in the **with Namespace** box.

Note that you must provide the Namespace ID of the namespace, not the descriptive name of the namespace.

5. Click **OK**.

The system connects you to the specified server, and adds the server name under the Replications icon  for the target server.

After you create a replication connection, you can manually synchronize data  between the target and source servers or create a chore to automate synchronization. For details, see [“The synchronization process” on page 114](#).

Modify a replication connection

You can change the user name and password properties of a replication connection. If you change these properties, you are logged in to the replication server as a different user. The existing replication relationships between the source cube and the mirror cube may no longer be valid. The privileges of the new client may not be equivalent to those of the old client. In this case, the synchronization process does not work.

Procedure

1. Open the Server Explorer.
2. Access the server that contains the replication connection you want to modify.

3. Double-click the Replications icon to expand it.
4. Right-click the server whose connection you want to modify, and click **Modify Replication Parameters**.

The Modify Server Replication Object dialog box opens.

5. Make the appropriate changes, and click **OK**.

Delete a replication connection

You can sever the relationship between the source and target servers by deleting a replication connection. When you do this, you delete the relationship between the source and mirror cubes. You can still update your local copy of the cube, but you can no longer synchronize the updates with the source cube.

Procedure

1. Open the Server Explorer.
2. Access the server that contains the replication connection you want to delete.
3. Double-click the Replications icon to expand it.
4. Right-click the server whose connection you want to delete and click **Delete Replication**.

Note: If you created a chore to automate synchronization, you must delete the chore before deleting the replication. For details on automating synchronization, see [“The synchronization process” on page 114](#).

Appendix A. Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products.

Keyboard shortcuts

Standard Microsoft Windows navigation keys are used in addition to application-specific keys.

You can use keyboard shortcuts to navigate through the application and perform tasks. If you are using a screen reader, you might want to maximize your window so the keyboard shortcut table is completely expanded and accessible.

Note: The following keyboard shortcuts are based on US standard keyboards.

Action	Shortcut key
Perform the command for an active command button	Enter
Close an object or pane that is closable	Ctrl+W
Go to the first item or object; Go Home	Home
Go to the last item or object; Go to the End	End
Move forward through the panes of the application	F8
Move backward through the panes of the application	Shift+F8
Move the focus to the Application Bar (blue dot)	Alt+F10
Move to the next item in the tab index order at the same level; cycle to the first tab index when you are at the end	Tab
Move to the previous item in the tab index order at the same level; cycle to the last tab index when you are at the beginning	Shift+Tab
Toggle on or off	Space bar
Move to the next option button and select it	Right arrow, Down arrow
Move to the previous option button and select it	Up arrow, Left arrow
Open and display the contents of a drop-down list	Down arrow
Close an open drop-down list	Esc
Move to the next selectable node after the current node. If the node that you select has children nodes and is expanded, go to the first child node	Down arrow
Move to the previous selectable node	Up arrow
Expand the current selection if it is not expanded. If the node is expanded, go to the first child node	Right arrow and plus sign

Table 5. Keyboard shortcuts (continued)

Action	Shortcut key
Collapse the current selection if it is expanded. If the node is collapsed, go to the parent node before the current selection	Left arrow and minus sign
Expand the children menu items	Right arrow
Collapse the children menu items	Left arrow
Open a context menu	Right-click key (Mozilla Firefox); Shift+F10 (Microsoft Internet Explorer)
Close an open context menu	Esc
Scroll down	Down arrow or Page Down
Scroll up	Up arrow or Page Up
Move to the next widget in the tab index order at the same level in the canvas	Tab
Move to the previous widget in the tab index order at the same level in the canvas	Shift+Tab
Add a database instance	Alt+N
Upload a configuration file	Alt+U
Download a configuration file	Alt+I
Tile vertically	Alt+Q
Tile horizontally	Alt+W
Box tile	Alt+P
Refresh the tree	Alt+R

IBM and accessibility

See the IBM Accessibility Center for more information about the commitment that IBM has to accessibility.

IBM Accessibility Center (<https://www.ibm.com/able/>)

Appendix B. Control Cubes

IBM TM1 uses control cubes.

The IBM TM1 server automatically generates the control cubes. By default, logging is enabled for all control cubes.

The following information is provided for each control cube:

- Purpose
- Dimensions that form the cube

Note: For a complete description of the control dimensions, see [Appendix C, “Control Dimensions,”](#) on page 135.

To toggle the display of control cubes and other control objects in Server Explorer, click **View, Display Control Objects**. All control cubes appear in Server Explorer with a prefix of a right curly brace (}). For example, }CellSecurity_SalesCube.

Security Control Cubes

Security control cubes apply security privileges for TM1 objects to user groups on the TM1 server.

Most of these control cubes are populated with the privileges assigned in the TM1 Security Assignments window, but you can also apply privileges directly in the control cubes. However, you cannot apply privileges to the ADMIN group; this group always has ADMIN privileges to all objects on the TM1 server.

}CellSecurity_CubeName

TM1 creates an empty }CellSecurity_CubeName control cube when you initiate the process of defining the cell-level security.

To do so, select a cube in Server Explorer, and click **Security, Create Cell Security Cube**. You can use the control cube to define the cell-level security for the selected cube. TM1 applies the cell-level security to the user groups.

The cells in a }CellSecurity_CubeName control cube can contain one of the following three strings, which correspond to the security privileges that you can assign to the cells:

- Read
- Write
- None

For details on defining the cell-level security and for definitions of the security privileges, see *IBM Cognos TM1 for Developers*

Dimensions

A }CellSecurity_CubeName cube contains all the dimensions that are present in the cube for which you want to define the cell-level security. In addition, a }CellSecurity_CubeName cube contains the following dimension:

Dimension	Elements
}Groups	Each user group defined for the TM1 server.

For example, if you create a cell security control cube for SalesCube in the TM1 sample data, the resulting }CellSecurity_SalesCube cube contains all the dimensions of SalesCube (actvsbud, region, model, account1, and month), and the }Groups dimension.

}ChoreSecurity

The }ChoreSecurity control cube stores security privileges for all chores on the TM1 server.

This control cube is populated with the values applied in the TM1 Security Assignments window, which you can access by clicking **Chores, Security Assignments** in Server Explorer.

You can also enter security privileges directly in the }ChoreSecurity cube.

The cells in the }ChoreSecurity cube can contain the string Read, which assigns the Read security privilege. The cells can also be left empty, which assigns the None security privilege.

For details on using the TM1 user interface to define chore security and for definitions of security privileges, see *TM1 for Developers*.

Dimensions

The }ChoreSecurity control cube contains the following dimensions:

Dimension	Elements
}Chores	Each chore currently defined on the TM1 server.
}Groups	Each user group defined for the TM1 server.

}ClientSecurity

The }ClientSecurity control cube defines user group access to properties for individual clients on the TM1 server.

For example, in the standard SData sample database that is included with the TM1 server, the ADMIN group has ADMIN access to properties for all clients on the server. The North America and South America groups have READ access to properties for all clients on the server.

For example, in the standard SData sample database that is included with the TM1 server, the ADMIN and DataAdmin groups have ADMIN access to properties for all clients on the server. The SecurityAdmin group has READ access to properties for all clients on the server, and the North America and South America groups have NONE access to properties.

The access privileges defined in the }ClientSecurity cube are important because several API functions require specific privileges to read and/or set client properties. For instance, the TM1ClientPasswordAssign function can be used only by clients who are members of a group with ADMIN access to properties, for the client to whom a password is assigned. Similarly, the TM1ObjectPropertyGet function requires READ access to client properties, while TM1ObjectPropertySet requires WRITE access.

TM1 does not populate this cube through the UI, nor is access assigned when you create a new user or new group. You must assign access privileges directly in the }ClientSecurity cube.

}CubeSecurity

The }CubeSecurity control cube stores security privileges for all cubes, including other control cubes on the TM1 server.

This control cube is populated with the values applied in the TM1 Security Assignments window, which you can access by clicking **Cubes, Security Assignments** in Server Explorer.

You can also enter security definitions directly in the }CubeSecurity cube.

The cells in the }CubeSecurity control cube can contain one of the following five strings, which correspond to the security privileges that you can assign to cubes:

- Read
- Write
- Reserve

- Admin
- Lock

The cube does not accept the string None. To assign the None privilege to a cube, leave the appropriate cell in the }CubeSecurity cube empty.

For details on defining cube security through the TM1 user interface and for definitions of security privileges, see the IBM Cognos *TM1 for Developers*.

Dimensions

The }CubeSecurity control cube contains the following dimensions:

Dimension	Elements
}Cubes	Cubes available on the TM1 server.
}Groups	Groups defined for the TM1 server.

}DimensionSecurity

The }DimensionSecurity control cube stores security definitions for all dimensions, including control dimensions on the TM1 server.

This control cube is populated with the values applied in the TM1 Security Assignments window, which you can access by clicking **Dimensions, Security Assignments** in Server Explorer.

You can also enter security definitions directly in the }DimensionSecurity cube.

The cells in the }DimensionSecurity control cube can contain one of the following five strings, which correspond to the security privileges that you can assign to dimensions:

- Read
- Write
- Reserve
- Admin
- Lock

The cube does not accept the string None. To assign the None privilege to a dimension, leave the appropriate cell in the }DimensionSecurity cube empty.

For details on defining dimension security through the TM1 user interface and for definitions of security privileges, see *TM1 for Developers*.

Dimensions

The }DimensionSecurity control cube contains the following dimensions:

Dimension	Elements
}Dimensions	Dimensions available on the TM1 server.
}Groups	Groups defined for the TM1 server.

}ElementSecurity_DimensionName

The }ElementSecurity_DimensionName control cube stores security definitions for all the elements in the DimensionName dimension.

A unique }ElementSecurity_DimensionName cube exists for each dimension for which you have defined element security.

This control cube is populated with the values applied in the TM1 Security Assignments window, which you can access by selecting a dimension and clicking **Dimension, Security, Elements Security Assignments** in Server Explorer.

You can also enter security definitions directly in an }ElementSecurity_DimensionName cube.

The cells in this control cube can contain one of the following five strings, which correspond to the security privileges that you can assign to dimensions:

- Read
- Write
- Reserve
- Admin
- Lock

The cube does not accept the string None. To assign the None privilege to an element, leave the appropriate cell in the control cube empty.

For details on defining element security through the TM1 user interface and for definitions of security privileges, see *IBM Cognos TM1 for Developers*.

Dimensions

An }ElementSecurity_DimensionName cube contains the following dimensions:

Dimension	Elements
DimensionName	All elements of the dimension are present.
}Groups	Groups defined for the TM1 server.

}ProcessSecurity

The }ProcessSecurity control cube stores security definitions for all TurboIntegrator processes on the TM1 server.

Dimensions

The }ProcessSecurity cube contains the following dimensions:

Dimension	Elements
}Processes	All processes defined on the TM1 server.
}Groups	Groups defined for the TM1 server.

Client and Group Administration Control Cubes

The client and group administration control cubes assign clients to user groups and store properties for all clients on the TM1 server.

}ClientsGroups

The }ClientsGroups cube stores group assignments for all clients on the TM1 server.

This control cube is populated with values that reflect the assignments applied in the Clients/Groups window, which you can access by clicking **Server, Security, Clients/Groups** in Server Explorer. You can also enter values directly in the }ClientsGroups cube.

For details on assigning clients to user groups, see *IBM Cognos TM1 for Developers*.

The cells in the }ClientsGroups cube contain strings that assign clients to one or more of the user groups available on the TM1 server.

In this example:

- Admin client is assigned to the ADMIN group
- Usr1 client is assigned to the North America group
- Usr2 client is assigned to the South America group
- Usr3 client is assigned to both the North America and South America groups

	}Groups				
}Clients	ADMIN	North America	South America	SecurityAdmin	DataAdmin
Admin	ADMIN				
Usr1		North America			
Usr2			South America		
Usr3		North America	South America		

Dimensions

The }ClientsGroups control cube contains the following dimensions:

Dimension	Elements
}Clients	Clients currently defined for the TM1 server.
}Groups	Groups defined for the TM1 server.

}ClientProperties

The }ClientProperties control cube stores property values for all clients on the TM1 server.

You can use client properties to define a client password, the maximum number of connections allowed, client status, and more.

The }ClientProperties cube is populated with values that reflect the settings in the Clients/Groups window, which you can access by clicking **Server, Security, Clients/Groups** from Server Explorer.

A MaximumPorts value of 0 for a client in the }ClientProperties cube indicates that a maximum port limit is not defined for the client.

You can enter or edit values for MaximumPorts and PasswordExpirationDays directly in the }ClientProperties cube.

Note: You should not edit values for PASSWORD directly in the cube. The passwords are stored in the cube in an encrypted format; if you edit passwords directly in the cube, the passwords are unusable and prevent clients from logging on to the TM1 server.

Dimensions

The }ClientProperties cube contains the following dimensions:

Dimension	Elements
}Clients	Clients currently defined for the TM1 server.
}ClientProperties	Properties that can be set for TM1 clients, including Password, PasswordExpirationDays, and MaximumPorts. For complete details on this dimension, see Appendix C, "Control Dimensions," on page 135.

Object Attribute and Property Control Cubes

The object attribute and property control cubes store attribute and property values for objects on the TM1 server.

}ConnectionProperties

The }ConnectionProperties control cube stores property values for all replication connections on the TM1 server.

The replication connection properties define the client and password to establish a replication connection, and the synchronization status for the star and planet servers.

Dimensions

The }ConnectionProperties cube contains the following dimensions:

Dimension	Elements
}Connections	Replication connections defined for the TM1 server.
}ConnectionProperties	Properties that can be set for replication connections, including User, Password, Syncstar and Syncplanet. For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 135.

}CubeProperties

The }CubeProperties control cube stores property values for all cubes, both native and replicated, on the TM1 server.

This control cube is populated with values that reflect the settings in the Cube Properties dialog box, which you can access by selecting a cube and clicking **Cube, Properties** in Server Explorer.

With replicated cubes, the }CubeProperties cube is populated with values that reflect the settings in the Replicate Cube dialog box, which you can access by selecting a cube under a replication connection and clicking **Cube, Replicate** from Server Explorer.

The }CubeProperties control cube also stores several properties that can only be set directly in the control cube, such as VMM and VMT.

The cube properties define how a cube is loaded on the TM1 server, which dimension to use as a measures dimension, which dimension to use as a time dimension, and whether cube logging is enabled. With replicated cubes, the values in the }CubeProperties cube also define the source cube for a replication, the replication status of a cube, and whether rules and views should be synchronized.

Dimensions

The }CubeProperties cube contains the following dimensions:

Dimension	Elements
}Cubes	Cubes available on the TM1 server.
}CubeProperties	Properties that can be set for cubes. For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 135.

}DimensionProperties

The }DimensionProperties control cube stores property values for all dimensions, both native and replicated, on the TM1 server.

This control cube is populated with values that reflect the settings in the DimensionElementOrdering dialog box, which you can access by selecting a dimension and clicking **Dimension, Set Elements Order** from Server Explorer.

With replicated cubes, the }DimensionProperties cube is also populated with values that reflect the settings in the Replicate Cube dialog box, which you can access by selecting a cube under a replication connection and clicking **Cube, Replicate** in Server Explorer.

The dimension properties define how a dimension is loaded and sorted on the TM1 server. With replicated dimensions, the values in the }DimensionProperties cube also define the source dimension for a replicated dimension, the replication status of a dimension, and whether subsets and attributes should be synchronized.

Dimensions

The }DimensionProperties cube contains the following dimensions:

Dimension	Elements
}Dimensions	Dimensions available on the TM1 server.
}DimensionProperties	Properties that can be set for dimensions. For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 135.

}DimensionAttributes

The }DimensionAttributes control cube stores attribute values for all dimensions, both native and replicated, on the IBM TM1 Server.

This control cube is populated with values that reflect the settings in the Attributes Editor window, which you can access by selecting a Dimensions group and clicking **Dimensions, Edit Attributes** from Server Explorer. You can also enter attribute values directly in the }DimensionAttributes cube.

Note: The }DimensionAttributes control cube is not created by default when you start the server. It is created when the server is accessed using IBM Planning Analytics Workspace or IBM TM1 Performance Modeler.

Dimensions

The }DimensionAttributes cube contains the following dimensions:

Dimension	Elements
}Dimensions	Dimensions available on the TM1 Server.
}DimensionAttributes	Attributes that have been defined for dimensions on the TM1 Server.

}ElementAttributes_DimensionName

The }ElementAttributes_DimensionName control cube stores element attribute values for the DimensionName dimension.

A unique }ElementAttributes_DimensionName cube exists for each dimension for which you have defined element attributes.

An }ElementAttributes_DimensionName cube is populated with values that reflect the settings in the Attributes Editor window, which you can access by selecting a dimension and clicking **Dimension, Edit Element Attributes** in Server Explorer. You can also enter element attribute values directly in an }ElementAttributes_DimensionName cube.

Dimensions

An }ElementAttributes_DimensionName cube contains the following dimensions:

Dimension	Elements
DimensionName	All elements of the dimension are present.
}ElementAttributes_DimensionName	Element attributes defined for the DimensionName dimension. For details on defining element attributes, see <i>IBM Cognos TM1 for Developers</i> .

}HierarchyProperties

The }HierarchyProperties control cube stores custom named levels for the hierarchy levels of TM1 dimensions.

You can enter your own names for these levels in the }HierarchyProperties control cube and then use these names outside of TM1 to access TM1 data with IBM Cognos Report Studio, MDX statements or other MDX OLAP tools. You can also use this control cube to assign a default member for the dimension.

For details on using named levels with dimensions, see the related section in *TM1 for Developers*.

Dimensions

The }HierarchyProperties cube contains the following dimensions:

Dimension	Elements
}Dimensions	Contains a list of all the dimension names in the TM1 server.
}Hierarchies	Includes only one element for assigning named levels to the selected dimension.
}HierarchyProperties	Includes the following two types of elements <ul style="list-style-type: none"> • A single element, named defaultMember, for assigning the default member of the dimension. • A set of 21 elements (level000 - level020) for assigning names to the dimension hierarchy levels. For more details, see “}HierarchyProperties” on page 140.

Performance Monitoring Control Cubes

TM1 includes a performance monitoring feature that lets you record performance statistics for clients, cubes, and servers.

When you enable performance monitoring, TM1 populates the performance monitoring control cubes when the server starts. You can browse these cubes to analyze server performance.

Note: TM1 Server reports the memory that is consumed by feeders only once. If there are no changes to rules or cube data, the memory consumption represents a static value that is recorded after the server

successfully starts. If there are changes to rules, feeders, or cube data during the running of the server, the Memory Used for Feeders value in the }StatsByCube control cube is updated the next time that the server starts.

You can use the `Maintain_detailed_feeder_memory_stats` cube property to generate statistics for detailed feeder memory usage for a specific cube. A value of 'YES' for the (`cubeName` , `propertyName`) tuple turns on the detailed feeder memory used calculation.

Note: The detailed feeder memory used calculation is intended to be used during pre-production as an aid to help design the cube. When this step is complete, the property value should be switched to 'NO' so that the much faster feeder memory used calculation is used during production.

Performance monitoring is enabled on a per-server basis. To enable performance monitoring for a server, select the server in Server Explorer and click **Server, Start Performance Monitor**. For more information, see Chapter 3, “Remote TM1 Server operations,” on page 13.

}StatsByClient

For each client on the server, the }StatsByClient control cube tracks the message count, average message size, total elapsed time, and other measures.

Dimensions

The }StatsByClient cube contains the following dimensions:

Dimension	Elements
}StatsStatsByClient	This dimension contains the following measures: Message Count Message Bytes Request Count Elapse Time Bytes/Message Measures are described in Appendix C, “Control Dimensions,” on page 135 .
}PerfClients	Each client on the server, plus the consolidated element Clients Total, which is a consolidation of all clients.
}TimeIntervals	See a description of the }TimeIntervals dimension in Appendix C, “Control Dimensions,” on page 135 .

}StatsByCube

For each cube on the server, the }StatsByCube control cube tracks memory use, the number of populated string cells, populated numeric cells, stored calculated cells, fed cells, stored views, and other statistics.

Dimensions

The }StatsByCube cube contains the following dimensions:

Dimension	Elements
}StatsStatsByCube	<p>This dimension contains the following measures:</p> <p>Memory Used for Calculations</p> <p>Memory Used for Feeders</p> <p>Memory Used for Input Data</p> <p>Memory Used for Views: Counts views that have been opened. Counts all views, whether private or public, regardless of user. This count is not dependent on who happens to be logged in.</p> <p>Number of Fed Cells</p> <p>Number of Populated Numeric Cells</p> <p>Number of Populated String Cells</p> <p>Number of Stored Calculated Cells</p> <p>Number of Stored Views: Counts all views that have been opened, whether private or public, regardless of user. Includes all views that have been looked at.</p> <p>Rule calculation cache miss rate: This measure reports cache efficiency for rules calculations, expressed as a percentage. A low cache miss rate indicates that cache is being used efficiently. As the reported cache miss rate increases, cache benefit diminishes. The cache miss rate is calculated with this expression: $100 * (M / (M + P))$. M is the count of cache cell retrieval attempts that missed. P is the total number of populated numeric leaf cells in the cube.</p> <p>Steps of Average Calculation: The average number of internal steps required to perform rule evaluation for the cube. This calculation can be directly compared to the CALCULATIONTHRESHOLD cube property that determines whether values are added to the rule calculation cache.</p> <p>Total Memory Used</p>
}PerfCubes	<p>This dimension contains numeric elements for each cube on the server, plus the consolidated element Cubes Total, which is a consolidation of all clients.</p>
}TimeIntervals	<p>See a description of the }TimeIntervals dimension in Appendix C, "Control Dimensions," on page 135.</p>

}StatsByCubeByClient

For each client and cube on the server, the }StatsByCubeByClient control cube tracks the number and elapsed time of cell updates, cell retrievals, view calculations, and view retrievals.

Cell updates do not work when setting cells in a view via Architect.

Dimensions

The }StatsByCubeByClient cube contains the following dimensions:

Dimension	Elements
}PerfClients	Each client on the server, plus the consolidated element Clients Total, which is a consolidation of all clients.
}PerfCubes	Each cube on the server, plus the consolidated element Cubes Total, which is a consolidation of all cubes.
}CubeFunctions	This dimension contains the following numeric elements: Cell Update Cell Retrieval View Calculation View Retrieval For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 135.
}StatsStatsByCubeByClient	This dimension contains the following measures: Count Elapse Time (milliseconds) For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 135.
}TimeIntervals	See a description of the }TimeIntervals dimension in Appendix C, “Control Dimensions,” on page 135.

}StatsByRule

The }StatsByRule control cube stores performance statistics for cube rules on the TM1 server.

To enable the storage of statistics for a rule, you must set the Rule_Stats property for the associated cube to YES in the [}CubeProperties control cube](#).

Each time a rule is changed or compiled, the data for that rule is cleared and updated in the }StatsByRule control cube. This helps you to immediately see the impact of a rule change.

The data in the }StatsByRule control cube does not persist between server sessions, it is cleared every time that you restart your TM1 server.

The }StatsByRule control cube contains the following dimensions:

Dimension	Elements
}Cubes	Each cube on the TM1 server is included as an element in the }Cubes dimension. To view statistics for a different cube rule, select a new cube name from the }Cubes dimension.
}LineNumber	Numeric elements 1 through 10,000. The elements correspond to line numbers in a TM1 rule. Note: The Rules Editor in TM1 Architect and Perspectives does not indicate line numbers. To view a rule with line numbers, open the <i>cube_name.rux</i> file in a text editor that includes line numbers. The Rules Editor in TM1 Performance Modeler does include line numbers.

Dimension	Elements
}RuleStats	<p>Contains elements that correspond to the information and statistics that are stored for rules on your TM1 server, including:</p> <ul style="list-style-type: none"> • Rule Text • Total Run Count • Min Time • Max Time • Avg Time • Total Time • Last Run Time <p>For details, see “}RuleStats” on page 142 control dimension.</p>

The times recorded for Min Time, Max Time, Avg Tme, Last Run time, and Total Time are in milliseconds (one one-thousandth of a second). Some rules execute faster than 1 millisecond, resulting in an entry of 0 for the time. It's possible for a simple rule to run multiple times, while the Total Time shows as 0.

}StatsForServer

The }StatsForServer control cube tracks the connected clients, active threads, and memory used for the server.

Dimensions

The }StatsForServer cube contains the following dimensions:

Dimension	Elements
}StatsStatsForServer	<p>This dimension contains the following measures:</p> <ul style="list-style-type: none"> Number of Connected Clients Number of Active Threads Memory Used (bytes) Memory in Garbage (bytes) <p>For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 135.</p>
}TimeIntervals	<p>See a description of the }TimeIntervals dimension in Appendix C, “Control Dimensions,” on page 135.</p>

Other Control Cubes

The following sections describe all other control cubes available on the TM1 server.

}Hold_UserName_CubeName

The }Hold_UserName_CubeName control cube tracks the cells held by the UserName client on the CubeName cube.

Dimensions

The }Hold_UserName_CubeName cube contains the following dimensions:

Dimension	Elements
}Hold	This dimension contains the following elements: OriginalVal Hold Status For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 135.

Other Dimensions in the }Hold_UserName_CubeName cube

The }Hold_UserName_CubeName cube also contains all of the dimensions present in the CubeName cube. For example, the }Hold_Admin_Sales cube contains the }Hold dimension as well as all the dimensions present in the Sales cube.

Appendix C. Control Dimensions

IBM TM1 uses control dimensions to track performance statistics, administer security, manage clients and groups, and store object attributes and properties.

}Chores

The }Chores dimension is used only in the }ChoreSecurity control cube.

The dimension contains string elements corresponding to the names of the chores defined on the IBM TM1 server.

}ClientProperties

This }ClientProperties dimension is used only in the }ClientProperties control cube.

The dimension contains the following string elements that corresponds to the following properties that can be defined for clients on the IBM TM1 server.

Element	Description
MaximumPorts	Indicates the maximum number of ports (connections) that an individual client can open on the TM1 server.
PASSWORD	Stores and displays the client password in an encrypted form.
PasswordExpirationDays	Indicates the number of days for which a given client's password remains valid.
PasswordLastTimeUpdated	Indicates the last time (GMT) the password for a given client was updated. The time format is yyymmddhhss.
STATUS	When a client is logged on to the TM1 server, the STATUS value is ACTIVE. Otherwise the STATUS value is empty.

}Clients

The }Clients dimension contains string elements that correspond to the clients defined on the IBM TM1 server.

This dimension is used in the }ClientGroups and }ClientProperties control cube.

}ConnectionProperties

The }ConnectionProperties dimension is used only in the }ConnectionProperties control cube.

This dimension contains the following string elements that correspond to the properties that can be defined for replication connections on the IBM TM1 server.

Element	Description
INTEGRATEDSECURITY	Determines if the replication connection uses Integrated Login. If the connection uses Integrated Login, the value of this property is YES, otherwise the value is NO.

Element	Description
LASTSYNC	Time (GMT) at which the data was last synchronized on the planet server, from which the replication connection was established. The time format is yyyyymmddhhss.
LASTSYNCSTAR	Time (GMT) at which the data was last synchronized on the star server for the replication connection. The time format is yyyyymmddhhss.
LASTSYNCSTARRECORD	Time (GMT) at which the most recent synchronization resulted in data moving from the star server to the planet. The time format is yyyyymmddhhss. This element differs from LASTSYNCSTAR in that LASTSYNCSTAR reports the time of the most recent synchronization, even if the synchronization did not result in any data moving between star server to the planet.
NAMESPACE	The namespace ID of the Cognos namespace when establishing a replication connection to a server that uses IBM Cognos 8 security. Note that this property stores the ID of the Cognos namespace, <i>not</i> the descriptive name of the namespace.
PASSWORD	Stores and displays the password in an encrypted format for establishing a replication connection.
SYNCPLANET	Determines if the data on the planet server is updated during the synchronization process. If the data on the planet server is updated, the value of this property is YES, otherwise the value is NO.
SYNCSTAR	Determines if the data on the star server is updated during the synchronization process. If the data on the star server is updated, the value of this property is YES, otherwise the value is NO.
USER	For a given replication connection, stores the name of the user who established the connection.

}Connections

The }Connections dimension contains string elements that correspond to the replication connections defined on the IBM TM1 server. This dimension is used only in the }ConnectionProperties control cube.

}CubeFunctions

The }CubeFunctions dimension is used only in the }StatsByCubeByClient control cube.

The dimension contains the following string elements corresponding to functions that can be performed on a cube:

Element	Description
Cell Retrieval	Cells retrieved by a user in a non-stored view. This metric includes calculated cells, such as rules, consolidations, and user-defined calculations.

Element	Description
Cell Update	Cell updates include user input through the Cube Viewer, In-Spreadsheet Browser, TM1 Web, and slices. Cells updated through rule calculations or other methods are not included in this metric.
View Calculation	Number of views requested, which can be user requests for Cube Viewer or slices.
View Retrieval	Number of views requested and displayed.

}CubeProperties

The }CubeProperties dimension is used only in the }CubeProperties control cube.

The dimension contains the following string elements corresponding to the properties that can be defined for cubes on the IBM TM1 Server.

Element	Description
CALCULATIONTHRESHOLD	Sets the threshold for the calculation cache in terms of number of calculation steps. Setting the cube property overrides the global value CalculationThresholdForStorage that is defined in the tm1s.cfg file.
DemandLoad	Indicates if a cube is automatically loaded when the server starts or is loaded 'on demand' only when a cube value is requested. When a cube is loaded on demand, the value of the DemandLoad property is YES, otherwise the property value is NO.
Lock	When a cube is locked, this property stores the name of the TM1 client that has locked the cube.
Logging	Indicates if logging is enabled for a given cube. When cube logging is enabled, the value of this property is YES, otherwise the value is NO.
Measures_Dimension	If a measures dimension has been defined for a given cube, this property stores the name of the dimension.
Replication	For replicated cubes, this property stores the name of the replication connection with which the replicated cube is associated.
RepSrcName	For replicated cubes, this property stores the name of the source cube.
RepStatus	If a cube is replicated on the server, the value of RepStatus is Copied, otherwise this property value is empty.

Element	Description
Rule_Stats	<p>This property determines if performance statistics are collected for the rules associated with the cube.</p> <p>To enable statistic collection, set the Rule_Stats property to YES. To disable statistic collection, set the property to NO (this is the default value).</p> <p>Rules statistics are stored in the }StatsByRule control cube.</p> <p>Rule_Stats is a dynamic property, meaning that it does not require a server restart to take effect. However, it can take up to 60 seconds for a dynamic property change to be applied on the TM1 server.</p>
SyncRule	For replicated cubes, this property indicates whether rules are synchronized when the associated cube is synchronized.
SyncViews	For replicated cubes, this property indicates whether views are synchronized when the associated cube is synchronized.
Time_Dimension	If a time dimension has been defined for a given cube, this property stores the name of the dimension.
VMM	<p>For each cube, this property determines the amount of RAM reserved on the server for the storage of stargate views. The more memory made available for stargate views, the better performance will be. You must, however, make sure sufficient memory is available for the TM1 server to load all cubes.</p> <p>The value of VMM is expressed in kilobytes. If no VMM value is specified the default value is 128 kilobytes.</p> <p>The valid range for VMM is 0 - 2,147,483,647 KB. The actual upper limit of VMM is determined by the amount of RAM available on your system.</p>
VMT	<p>For each cube, this property defines the time threshold, in seconds, beyond which the algorithm that stores TM1 stargate views is triggered.</p> <p>If the time required to calculate a cube view surpasses the specified threshold, TM1 attempts to store a stargate view. If there is not enough memory available to store the stargate view, TM1 purges the oldest stargate view that is not currently in use, and continues to purge views in this manner until sufficient memory is made available.</p> <p>If no VMM value is specified the default value is five seconds. The valid range is 1 - 259,200 seconds.</p> <p>For details on stargate views, see “About Stargate views” on page 21.</p>

}Cubes

The }Cubes dimension contains string elements corresponding to the cubes, including control cubes, on the IBM TM1 server.

}DimensionAttributes

The }DimensionAttributes dimension contains numeric elements that correspond to the attributes defined for dimensions on the IBM TM1 Server.

The }DimensionAttributes dimension is used in the }DimensionAttributes control cube.

Note: The }DimensionAttributes dimension is not created by default when you start the server. It is created when the server is accessed using IBM Planning Analytics Workspace or IBM TM1 Performance Modeler.

}DimensionProperties

The }DimensionProperties dimension is used only in the }DimensionProperties control cube.

The }DimensionProperties dimension contains the following string elements that correspond to the properties that can be defined for dimensions on the IBM TM1 server.

Element	Description
Default_Hierarchy	Not currently used.
DemandLoad	Not currently used.
Last_Time_Updated	Time (GMT) at which a dimension was last updated. The time format is yyyyymmddhhss.
Lock	When a dimension is locked, this property stores the name of the TM1 client that has locked the dimension.
Replication	For replicated cubes, this property stores the name of the replication connection with which the replicated cube is associated.
RepSrcName	For replicated dimensions, this property stores the name of the source dimension.
RepStatus	If a dimension is replicated on the server, the value of RepStatus is Copied. Otherwise the RepStatus value is empty.
SortComponentsSense	When the immediate components (children) of a consolidation are sorted, this property stores the sense applied to the sorting. Components can be sorted in either the Ascending or Descending sense.
SortComponentsType	This property indicates the type of sorting that is applied to the immediate components (children) of a consolidation. There are two types of sorting: ByName and ByInput. The ByInput sort does not actually enforce any sorting, it leaves the components in the order they appeared the last time the dimension was saved.
SortElementsSense	When dimension elements are sorted automatically, this property stores the sense applied to the sorting. Elements can be sorted in either the Ascending or Descending sense.

Element	Description
SortElementsType	When dimension elements are sorted automatically, this property stores the type of sorting used. There are four types of element sorting: ByName, ByLevel, ByHierarchy, and ByInput. The ByInput sort does not actually enforce any sorting, it leaves the elements in the order they appeared the last time the dimension was saved. If left blank, ByInput applies. SortElementsType applies to all dimension elements, both consolidations and leaf elements.
SyncAttributes	For replicated cubes, this property indicates whether element attributes are synchronized when the associated dimension is synchronized.
SyncSubsets	For replicated cubes, this property indicates whether subsets are synchronized when the associated dimension is synchronized.

}Dimensions

The }Dimensions dimension contains string elements that correspond to the dimensions, including control dimensions, on the IBM TM1 server.

The }Dimensions dimension is used only in the }DimensionsProperties control cube.

}ElementAttributes_DimensionName

The }ElementAttributes_DimensionName dimension contains numeric elements that correspond to the element attributes defined for the DimensionName dimension.

The }ElementAttributes_DimensionName dimension is used only in the }ElementAttributes_DimensionName control cube.

}Groups

The }Groups dimension contains string elements that correspond to the user groups defined on the IBM TM1 server.

The }Groups dimension is used only in the }ClientGroups control cube.

}Hierarchies

The }Hierarchies dimension is used in the }HierarchyProperties control cube.

This dimension contains only one element, named hierarchy0, that is used with the }HierarchyProperties dimension to assign named levels to the hierarchy levels of a TM1 dimension.

}HierarchyProperties

The }HierarchyProperties dimension is used in the }HierarchyProperties control cube, along with the }Hierarchies dimension, to assign named levels to the hierarchy levels of a TM1 dimension.

The }HierarchyProperties dimension contains the following elements:

Element	Description
defaultMember	<p>A string element that stores the name of the default member of the dimension.</p> <p>The element name you enter here may filter the dimension when TM1 data is retrieved from an external application like IBM Cognos Report Studio.</p> <p>Use the name of the top element in the dimension hierarchy so all the dimension elements are retrieved by default.</p>
level000 - level020	A set of string elements that store custom names for the hierarchy levels of a dimension.

}Hold

The }Hold dimension is used in any }Hold_UserName_CubeName cubes.

UserName is the name of the TM1 user who has applied holds against cube values, while *CubeName* is the name of the cube in which values are being held.

The }Hold dimension contains the following elements:

Element	Description
OriginalVal	This numeric element tracks the most recent value of leaf cells that are edited in the CubeName cube. These most recent values are used for several data spreading methods and should not be edited.
HoldStatus	<p>This string element identifies cells held by UserName in the CubeName cube.</p> <p>A value of H indicates that UserName has applied a leaf hold to a cell. A value of C indicates that UserName has applied a consolidation hold against a cell.</p>

}LineNumber

The }LineNumber control dimension contains numeric elements 1 through 10,000.

The elements in this control dimension correspond to the line numbers in a TM1 rules (.rux) file.

The }LineNumber control dimension is used exclusively in the }StatsByRule control cube, which stores information and statistics about your TM1 rules.

}PerfClients

The }PerfClients dimension contains numeric elements that correspond to the clients defined on the IBM TM1 Server as well as a consolidation of all clients named Clients Total.

The }PerfClients dimension is used in the following performance monitoring cubes:

- }StatsbyClient
- }StatsByCube
- }StatsByCubeByClient

The }PerfClients dimension differs from the }Clients dimension in that the }PerfClients dimension contains numeric elements, while the }Clients dimension contains string elements.

}PerfCubes

The }PerfCubes dimension contains numeric elements that correspond to the cubes on the IBM TM1 Server and a consolidation of all cubes named Cubes Total.

The }PerfCubes dimension is used in the }StatsByCube and }StatsByCubeByClient performance monitoring cubes.

The }PerfCubes dimension differs from the }Cubes dimension in that the }PerfCubes dimension contains numeric elements, while the }Cubes dimension contains string elements.

}Processes

The }Processes dimension contains string elements that correspond to the TurboIntegrator processes defined on the IBM TM1 server.

The }Processes dimension is used only in the }ProcessSecurity control cube.

}RuleStats

The }RuleStats control dimension contains elements that correspond to the information and statistics that are stored for rules on your TM1 server.

This control dimension is used exclusively in the }StatsByRule control cube, which stores information and statistics about the rules on your TM1 server.

The following elements are included in the }RuleStats control dimension.

Element	Description
Rule Text	The beginning of the rule text, to help you identify the rule.
Total Run Count	The total number of time the rule has been run.
Min Time	The minimum amount of time taken for the rule to run, in milliseconds.
Max Time	The maximum amount of time taken for the rule to run, in milliseconds.
Avg Time	The average rule run time, in milliseconds.
Total Time	The total amount of time taken by the rule, in milliseconds.
Last Run Time	The amount of time, in milliseconds, it took for the most recent execution of the rule.

}StatsStatsByClient

The }StatsStatsByClient dimension is used in the }StatsByClient control cube.

This dimension contains the following measures:

Element	Description
Bytes/Message	Average number of bytes per message. The value for this element is calculated by the statement ['Bytes/Message']=['Message Bytes']/['Message Count'] in the rule for the }StatsByClient cube.
Elapse Time (ms)	Time (in milliseconds) required to process requests to the TM1 server.
Message Bytes	Number of bytes that have been sent across in the requests.

Element	Description
Message Count	A message is a request to the TM1 server for a list of objects, such as a list of dimensions.
Request Count	Measures the number of requests for information from the TM1 server.

}StatsStatsByCube

The }StatsStatsByCube dimension is used only in the }StatsByCube control cube.

This dimension tracks performance statistics when performance monitoring is enabled, and contains the following elements:

Element	Description
Memory Used for Calculations	This metric measures the memory used in all TM1 calculations, including consolidations, rules, and user-defined calculations.
Memory Used for Feeders	This metric measures the amount of memory used to feed cells through TM1 rules.
Memory Used for Input Data	This includes memory allocated to data loaded through TurboIntegrator or process worksheets. This measures the memory used for all data input other than slice or Cube Viewer input.
Memory Used for Views	This metric measured the amount of memory, in bytes, used to store views for a given cube on the TM1 server.
Number of Fed Cells	This is the number of cells in the cube targeted by feeders.
Number of Populated Numeric Cells	This metric counts all the populated string cells in a given cube. String cells are cells identified by at least one string element.
Number of Populated String Cells	This metric counts all the populated string cells in a given cube. String cells are cells identified by at least one string element.
Number of Stored Calculated Cells	This metric counts all stored calculated cells including rules, dimensional consolidations, and user-defined consolidations.
Number of Stored Views	This metric measures the number of named views of a given cube.
Rule calculation cache miss rate	This measure reports cache efficiency for rules calculations, expressed as a percentage. A low cache miss rate indicates that cache is being used efficiently. As the reported cache miss rate increases, cache benefit diminishes.
Steps of Average Calculation	The average number of internal steps required to perform rule evaluation for the cube. This calculation can be directly compared to the CALCULATIONTHRESHOLD cube property that determines whether values are added to the rule calculation cache.
Total Memory Used	The total amount of memory used by a cube, measured in bytes.

}StatsStatsByCubeByClient

The }StatsStatsByCubeByClient dimension is used only in the }StatsByCubeByClient control cube.

The dimension contains the following elements:

Element	Description
Count	For a given cube function, this metric measures the number of functions executed.
Elapsed Time (ms)	For a given type of cube function, this metric measures the elapsed time required to execute all functions.

}StatsStatsForServer

The }StatsStatsForServer dimension is used only in the }StatsForServer control cube.

The dimension tracks performance statistics when performance monitoring is enabled, and contains the following elements:

Element	Description
Memory In Garbage	TM1 does not release memory back to the operating system while the TM1 server is running. Instead, the server retains unused memory in 'garbage' for later use. Memory that has been allocated to the server but is not in use is represented by this metric.
Memory Used	The total amount of memory used by the server, in bytes.
Number of Active Threads	This measures the number of threads active on the TM1 server. Each client connection constitutes a thread. Threads are also used to execute chores and processes.
Number of Connected Clients	This statistic measures the number of active connections to the TM1 server. An individual client can have multiple connections to the TM1 server active simultaneously. Each connection is counted in this metric.

}Subsets_<DimensionName>

The }Subsets_<DimensionName> dimension is used to track the subsets in each dimension. Each element in this dimension represents a subset.

The }Subsets_<DimensionName> dimension is dynamically and automatically created and destroyed when dimensions are created and destroyed. It is updated with elements when subsets are created and destroyed.

You can use this dimension to write TurboIntegrator processes that iterate over a list of subsets in a dimension. Attributes for this dimension can be used for localization of subsets names.

When you create or destroy a subset, the server writes to this dimension. This can cause locking in cases where two or more subsets are being created at the same time in different TurboIntegrator processes.

To avoid this locking pattern, you should create temporary subsets when it is possible, for example, when you don't need the object to exist after the TurboIntegrator process completes. Creating temporary subsets does not update the }Subsets_<DimensionName> dimension. For more information, see [SubsetCreate](#) in *TM1 Reference*.

}TimeIntervals

All performance monitoring control cubes use the }TimeIntervals control dimension.

This dimension contains 168 numeric elements for time intervals at the minute and hour level, as follows:

- 120 elements corresponding to the minutes of the current and the prior hour. Elements in the current hour are named 0M00, 0M01, ... 0M59. Elements in the prior hour are named 1M00, 1M01, ... 1M59.

Sampling for the performance monitoring cubes is done every minute. New values are stored in consecutive minutes of the current hour. When the hour is complete, the values of the current hour are copied to the prior hour, and the current hour is cleared to accept new values.

- 48 elements corresponding to the hours of the current and the prior day. Elements in the current day are named 0H00, 0H01, ... 0H23. Elements for the previous day are named 1H00, ... 1H01, 1H23. Every hour, a new summary (average) value from the minutes is posted in the corresponding hourly element. When the day is over, the current day's data replaces that of the prior day and the current day is cleared to accept new data.

Hours and minutes in this dimension reflect the system clock time when you started the performance monitor. For example, if you start the performance monitor at 10:31 AM local system time, the first }TimeIntervals element to receive a value is 0M31, which is the 31st minute of the current hour. The next element to receive a value would be 0M32.

}Views_CubeName

The }Views_<CubeName> dimension is used to track the views in each cube. Each element in this dimension represents a view.

The }Views_<CubeName> dimension is dynamically and automatically created and destroyed when cubes are created and destroyed. It is updated with elements when views are created and destroyed.

You can use this dimension to write TurboIntegrator processes that iterate over a list of views in a cube. Attributes for this dimension can be used for localization of view names.

When you create or destroy a view, the server writes to this dimension. This can cause locking in cases where two or more views are being created at the same time in different TurboIntegrator processes.

To avoid this locking pattern, you should create temporary views when it is possible, for example, when you don't need the object to exist after the TurboIntegrator process completes. Creating temporary views does not update the }Views_<CubeName> dimension. For more information, see [ViewCreate](#) in *TM1 Reference*.

Appendix D. Excel events handled by TM1

Some Microsoft Excel events are handled by the IBM TM1 add-in for Excel.

Excel Event Name	Description
NewWorkbook	This event is handled to track the number of workbooks opened.
WorkbookOpen	This event is handled to track the number of workbooks opened and also to set the state of TM1 toolbars and charts.
WorkbookBeforeSave	This event is handled to detect modified status of dependent TM1 Objects and to prompt save on such objects.
WorkbookBeforeClose	This event is handled to detect modified status of dependent objects and prompt save on such objects, as well as to unload TM1 toolbars and menu bars when the TM1 add-in is unloaded. To achieve this, TM1 detects the workbook saved status, posts an appropriate Save Prompt message box, and handles the saving of the current workbook.
WorkbookActivate	This event is handled to correctly set the status of TM1 toolbars and menu bars.
WorkbookAddinUninstall	This event is handled to remove TM1 toolbars and menu bars from a workbook.
SheetActivate	This event is handled to update the TM1 toolbar and menu bar states, as well as to update sheet-specific variables.
SheetDeactivate	This event is handled to detect sheet deletion.
SheetSelectionChange	This event is handled to update TM1 menus and toolbar states.
SheetChange	This event is handled to update the TM1 database with the changes made to a particular cell.
SheetBeforeDoubleClick	This event is handled to post TM1 specific dialog boxes on particular TM1 cells.
SheetBeforeRightClick	This event is handled to add and modify Excel context menu.

Notices

This information was developed for products and services offered worldwide.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. This document may describe products, services, or features that are not included in the Program or license entitlement that you have purchased.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
Attention: Licensing

3755 Riverside Dr.
Ottawa, ON
K1V 1B7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information here is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.



Product Information

This document applies to IBM Planning Analytics version 2.0.0 and may also apply to subsequent releases.

Copyright

Licensed Materials - Property of IBM

© Copyright IBM Corp. 2007, 2020.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web in "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Red Hat®, JBoss®, OpenShift®, Fedora®, Hibernate®, Ansible®, CloudForms®, RHCA®, RHCE®, RHCSA®, Ceph®, and Gluster® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Microsoft product screen shot(s) used with permission from Microsoft.

Index

Special Characters

- h [10](#)
- .cfg file [9](#), [12](#)
- .ini file [18](#)
- .rej file [13](#), [46](#)
- }CellSecurity_CubeName control cube [121](#)
- }Chores control dimension [135](#)
- }ChoreSecurity control cube [122](#)
- }ClientProperties control cube [125](#)
- }ClientProperties control dimension [135](#)
- }Clients control dimension [135](#)
- }ClientSecurity control cube [122](#)
- }ClientsGroups control cube [124](#)
- }ConnectionProperties control cube [126](#)
- }ConnectionProperties control dimension [135](#)
- }Connections control dimension [136](#)
- }CubeFunctions control dimension [136](#)
- }CubeProperties control cube [126](#)
- }CubeProperties control dimension [137](#)
- }Cubes control dimension [138](#)
- }CubeSecurity control cube [122](#)
- }DimensionAttributes control cube [127](#)
- }DimensionAttributes control dimension [139](#)
- }DimensionProperties control cube [127](#)
- }DimensionProperties control dimension [139](#)
- }Dimensions control dimension [140](#)
- }DimensionSecurity control cube [123](#)
- }ElementAttributes_DimensionName control cube [127](#)
- }ElementAttributes_DimensionName control dimension [140](#)
- }ElementSecurity_DimensionName control cube [123](#)
- }Groups control dimension [140](#)
- }Hierarchies control dimension [140](#)
- }HierarchyProperties control cube [128](#)
- }HierarchyProperties control dimension [140](#)
- }Hold control dimension [141](#)
- }Hold_UserName_CubeName control cube [132](#)
- }LineNumber control dimension [141](#)
- }PerfClients control dimension [141](#)
- }PerfCubes control dimension [142](#)
- }Processes control dimension [142](#)
- }ProcessSecurity control cube [124](#)
- }RuleStats control dimension [142](#)
- }StatsByClient [59](#)
- }StatsByClient control cube [129](#)
- }StatsByCube [59](#), [129](#)
- }StatsByCube control cube [129](#)
- }StatsByCubeByClient [59](#)
- }StatsByCubeByClient control cube [130](#)
- }StatsByRule control cube [131](#)
- }StatsForServer [59](#)
- }StatsForServer control cube [132](#)
- }StatsStatsByClient control dimension [142](#)
- }StatsStatsByCube control dimension [143](#)
- }StatsStatsByCubeByClient control dimension [144](#)
- }StatsStatsForServer control dimension [144](#)
- }Subsets_DimensionName control dimension [144](#)

- }TimeIntervals control dimension [145](#)
- }Views_CubeName control dimension [145](#)

A

- access
 - control [25](#)
 - privileges, and replication [110](#)
 - privileges, groups [27](#)
 - privileges, Read [110](#), [114](#)
 - privileges, synchronization [114](#)
 - privileges, Write [114](#)
- accessibility
 - keyboard shortcuts [119](#)
- add a server [67](#)
- Add Counters [62](#)
- ADMIN group
 - assigning passwords [34](#)
- Admin Host
 - location [12](#)
 - resetting [19](#)
- Admin server
 - diagram [9](#)
 - IP address [9](#)
 - name [9](#)
 - port number [9](#)
 - protocol [9](#)
 - UNIX [11](#)
 - Windows [10](#)
 - Windows Startup directory [10](#)
- Adminhost [67](#)
- AdminHost parameter
 - specify location [12](#)
- administrative groups [28](#)
- Allow Export as Text [3](#)
- Applications Server [73](#)
- assigning users to groups [33](#)
- Audit log
 - configuring [52](#)
 - copying data [55](#)
 - details [55](#)
 - exporting results [55](#)
 - find feature [55](#)
 - object events [51](#)
 - queries [53](#)
 - results [54](#)
 - system-wide events [51](#)
 - understanding events [51](#)
 - updating with latest records [52](#)
 - using [50](#)
 - viewing [53](#)
- Audit log, overview [50](#)
- authentication
 - integrated login [25](#)
- authentication, LDAP [25](#)

B

background mode [17](#)
Basic [69](#)
Block Access to Server Capability Assignment [3](#)
browser view [21](#)
bulk load mode [22](#)

C

cancel TM1 Top [89](#)
Capability assignments [3](#)
cell-level security [27](#)
chain configuration
 server [108](#)
change password [35](#)
chore
 synchronization [115](#)
clearing passwords [35](#)
client
 connections [20](#)
 disconnecting [20](#)
 messages [20](#)
 query [45](#)
client logging [100](#)
command prompt
 TM1 Top [88](#)
commit TM1 Top [87](#)
configure
 TM1 Top utility [82](#)
control cube
 }CellSecurity_CubeName [121](#)
 }ChoreSecurity [122](#)
 }ClientProperties [125](#)
 }ClientSecurity [122](#)
 }ClientsGroups [124](#)
 }ConnectionProperties [126](#)
 }CubeProperties [126](#)
 }CubeSecurity [122](#)
 }DimensionAttributes [127](#)
 }DimensionProperties [127](#)
 }DimensionSecurity [123](#)
 }ElementAttributes_DimensionName [127](#)
 }ElementSecurity_DimensionName [123](#)
 }HierarchyProperties [128](#)
 }Hold_UserName_CubeName [132](#)
 }ProcessSecurity [124](#)
 }StatsByClient [129](#)
 }StatsByCube [129](#)
 }StatsByCubeByClient [130](#)
 }StatsByRule [131](#)
 }StatsForServer [132](#)
 client and group administration [124](#)
 object attribute and property [126](#)
 performance monitoring [128](#)
 security [121](#)
control dimension
 }Chores [135](#)
 }ClientProperties [135](#)
 }Clients [135](#)
 }ConnectionProperties [135](#)
 }Connections [136](#)
 }CubeFunctions [136](#)
 }CubeProperties [137](#)

control dimension (*continued*)
 }Cubes [138](#)
 }DimensionAttributes [139](#)
 }DimensionProperties [139](#)
 }Dimensions [140](#)
 }ElementAttributes_DimensionName [140](#)
 }Groups [140](#)
 }Hierarchies [140](#)
 }HierarchyProperties [140](#)
 }Hold [141](#)
 }LineNumber [141](#)
 }PerfClients [141](#)
 }PerfCubes [142](#)
 }Processes [142](#)
 }RuleStats [142](#)
 }StatsStatsByClient [142](#)
 }StatsStatsByCube [143](#)
 }StatsStatsByCubeByClient [144](#)
 }StatsStatsForServer [144](#)
 }Subsets_DimensionName [144](#)
 }TimeIntervals [145](#)
 }Views_CubeName [145](#)
controlling object security [25](#)
copy
 dimension [113](#)
Counter Name [62](#)
cube
 enabling and disabling logging [45](#)
 mirror [108](#)
 replicating [107](#), [112](#)
 security [27](#)
 source [108](#), [112–115](#), [117](#)
 specifying replication information [113](#)
 synchronizing [107](#), [114](#)
Cube Dependency [103](#)

D

data
 backup [46](#)
data directory
 restricting access [34](#)
Data Reservation
 capabilities [3](#)
data, recovery [46](#)
DataAdmin group [30](#)
Date and Time [49](#)
DEBUG
 Admin server message severity [43](#)
 logging system message severity [49](#)
Dependency in Cubes [103](#)
dimension
 local [113](#)
 rename [113](#)
 security [27](#)
 specifying replication information [113](#)
dimension, copy [113](#)
DisableSandboxing [3](#)
disconnecting clients [20](#)
Display user names [68](#), [82](#)
Do Not Copy Rule [113](#)
download configuration [77](#)
download logs [77](#)

E

- element security [27](#)
- End Time [45](#)
- Enhanced [69](#)
- ERROR
 - Admin server message severity [43](#)
 - logging system message severity [49](#)
- exit TM1 Top [88](#)

F

- FATAL logging system message severity [49](#)
- File [39](#)
- Filtering [76](#)
- Format [39](#)

G

- garbage memory [21](#)
- group
 - adding [27](#), [33](#)
 - admin [29](#), [34](#)
 - multiple [34](#)

H

- health status [68](#)
- heartbeat interval [9](#), [10](#)

I

- IBM Cognos Performance Management Hub [80](#)
- idle
 - TM1 Top [87](#)
- INFO
 - Admin server message severity [43](#)
 - logging system message severity [49](#)
- integrated login [25](#)
- IX-lock [87](#)

K

- keyboard shortcuts
 - accessibility [119](#)

L

- LDAP
 - authentication [25](#)
- local dimension [113](#)
- local server [12](#), [19](#)
- lock modes [87](#)
- log
 - backing out transactions [47](#)
 - file [13](#), [45](#)
 - file, removing [47](#)
 - file, sample [19](#), [44](#)
 - tm1s.log file [44](#)
 - viewing [45](#)
- Log to disk [76](#), [77](#)
- Log_to_Disk [77](#)

- log, message [48](#)
- LogAppend [82](#)
- logging
 - disabling [45](#)
 - sample properties file [38](#)
 - system message severity [49](#)
 - transactions [19](#), [44](#)
- Logging [75](#)
- Logging in [67](#)
- logging, enabling [45](#)
- LogPeriod [82](#)
- LogReleaseLineCount [46](#)

M

- MaskUserNameInServerTools parameter [68](#), [82](#)
- MaxBackupIndex [39](#)
- MaxFileSize [39](#)
- memory
 - garbage [21](#)
 - management [21](#)
- memory usage graph [71](#)
- MemorySize [39](#)
- message
 - clients [20](#)
 - log [48](#)
 - text [49](#)
- Message Level [49](#)
- Message log
 - configuring [49](#)
 - properties file [38](#)
 - severity Levels [49](#)
 - using [48](#)
- metadata [13](#)
- mirror cube
 - definition [108](#)
 - dimension [113](#)
 - naming [113](#)
 - rules [112](#), [113](#)
 - specify information [112](#)
 - synchronization [114](#), [115](#)
- monitoring chores in TM1 Operations Console [71](#)
- MTQ
 - overview [2](#)
- multi-threaded queries, *See* MTQ

N

- numeric parameters [74](#)

O

- object security [25](#)
- operation group [67](#)
- Operations Console [73](#)
- out-of-process server [107](#)
- override dimension defaults [113](#)

P

- passwords
 - assigning [34](#)
 - for the ADMIN ID [18](#)

- passwords, admin group [34](#)
- passwords, change [35](#)
- passwords, clearing [35](#)
- passwords, setting [34](#)
- performance counters [60](#)
- performance monitoring
 - }StatsByClient [129](#)
 - }StatsByCube [129](#)
 - }StatsForServer [132](#)
 - control cubes [59](#), [128](#)
- Performance Statistics [70](#)
- Persistent feeders [6](#)
- personal workspace
 - managing [7](#)
- Personal Workspace Writeback Mode [3](#)
- PMHub, *See* IBM Cognos Performance Management Hub
- privilege
 - read access [110](#), [114](#)
 - write access [114](#)

R

- R-lock [87](#)
- read access privilege [110](#), [114](#)
- remote server
 - Admin host [12](#)
 - and replication [108](#)
 - application [13](#)
 - connecting [18](#)
 - diagram [13](#)
 - disconnecting [19](#)
 - memory management [21](#)
 - operations [13](#)
 - refreshing servers [18](#)
 - replication [108](#)
 - source server [108](#), [117](#)
 - starting [14](#)
 - target server [108](#), [117](#)
 - Windows service [15](#)
- removing log files [47](#)
- rename a dimension [113](#)
- replicate
 - a cube [108](#), [112](#)
 - a dimension [112](#), [113](#)
 - Admin Host considerations [111](#)
 - advantages [107](#)
 - connection [108](#), [116](#)
 - dimension name [112](#)
 - mirror cube name [112](#)
 - outcome [114](#)
 - overview [107](#)
 - process [112](#)
 - required access privileges [110](#)
 - server configurations [108](#)
 - server relationships [108](#)
 - setting up [111](#)
 - source server [108](#)
 - specifying rule information [113](#)
 - specifying cube information [112](#), [113](#)
 - specifying dimension information [113](#)
 - subset [114](#)
 - target server [108](#)
 - views [113](#)
- replication connection

- replication connection (*continued*)
 - creating [116](#)
 - deleting [117](#)
 - modifying [116](#)
 - overview [108](#)
- rollback TM1 Top [87](#)
- rules [112](#), [113](#)
- RunningInBackground [17](#)

S

- sandbox
 - behavior in different clients [8](#)
 - configuration parameters [7](#)
 - delete sandboxes [7](#)
 - files and folders [7](#)
 - managing [7](#)
 - memory usage [7](#)
- Sandbox Capability Assignment [3](#)
- sandbox queue [70](#)
- sandboxes, capability assignments [3](#)
- saving data [13](#)
- security
 - access privileges [27](#)
 - adding groups [33](#)
 - adding users [32](#)
 - assigning passwords [34](#)
 - change password [35](#)
 - clearing passwords [35](#)
 - cube-level [27](#)
 - dimension-level [27](#)
 - element-level [27](#)
 - group [27](#), [29](#), [33](#), [34](#)
 - multiple groups [34](#)
 - password security [34](#)
 - setting passwords [34](#)
 - TM1 data [34](#)
 - TM1 objects [25](#)
- SecurityAdmin group [29](#)
- server
 - Admin [9](#)
 - audit log [50](#)
 - local [12](#), [19](#)
 - out-of-process [107](#)
 - performance [19](#), [59](#)
 - performance monitoring [19](#), [59](#)
 - remote [12–14](#), [18](#), [19](#), [21](#), [108](#), [117](#)
 - remote shutdown [21](#)
 - remote shutting down [14](#)
 - source [108](#), [117](#)
 - star configuration [108](#)
 - start and chain configuration [108](#)
 - target [108](#), [117](#)
- server, message log [48](#)
- ServerName [82](#)
- service automatic start [15](#)
- Set Rule to Synchronize [113](#)
- setting passwords [34](#)
- shut down [49](#)
- shutdown_tm1s.sh [17](#)
- source cube [108](#), [112–115](#), [117](#)
- source server [108](#), [117](#)
- star and chain configuration [108](#)
- star configuration [108](#)

- Stargate view [21](#)
- Starting Time [45](#)
- startup_tm1s.sh [16](#)
- Status [69](#)
- synchronization
 - automatic [115](#)
 - on-demand [107](#), [115](#)
 - over unstable networks [116](#)
 - overview [108](#), [114](#)
 - required access privileges [114](#)

T

- target server [108](#), [117](#)
- thread details graph [72](#)
- Thread ID [49](#)
- TimeZone [39](#)
- TM1
 - log file [45](#)
 - Logger Name [49](#)
 - securing data [34](#)
 - transaction log file [19](#), [44](#)
 - users and groups security [27](#)
- TM1 Applications Server [72](#)
- TM1 Applications server logging [73](#)
- TM1 Operations Console [67](#), [79](#)
- TM1 PerfMon [60](#), [61](#)
- TM1 server
 - remote [18](#)
 - shutting down UNIX [17](#)
 - UNIX [16](#)
 - UNIX background mode [17](#)
 - Windows shutting down [14](#)
- TM1 Top utility
 - canceling threads [89](#)
 - command prompt [88](#)
 - configuring [82](#)
 - keys [82](#)
 - lock modes [87](#)
 - overview [81](#)
 - parameters [82](#)
 - thread processing states [87](#)
- tm1admsrv-log.properties [42](#)
- tm1admsrv-log.properties sample [42](#)
- Tm1admsrv.exe [10](#)
- tm1opsconsoleconfig.xml [77](#)
- Tm1p.ini file [18](#)
- tm1s_start_example [16](#)
- tm1s.cfg file [9](#)
- Tm1s.cfg file [12](#)
- tm1s.log
 - properties [38](#)
 - removing old log files [47](#)
- tm1srvstop.exe [14](#), [17](#)
- Tm1top.ini [82](#)
- tm1xfer [94](#)
- top utility [81](#)
- transaction log
 - query result [45](#)
- Transaction log [78](#)
- transaction log file [13](#)
- transaction log parameters
 - Cubes [45](#)
 - Starting Time [45](#)

- transaction log parameters, Clients [45](#)
- transferring servers on Windows and UNIX [94](#)
- TurboIntegrator Process monitoring [70](#)

U

- UNIX [16](#)
- upload configuration [77](#)
- user
 - adding [32](#)
- user and group security [27](#)

V

- value [62](#)
- Verify admin access [68](#)
- verify TM1 Top [88](#)
- View log [76](#)
- view, Stargate [21](#)

W

- W-lock [87](#)
- WARN logging system message severity [49](#)
- Watchdog [79](#)
- Windows
 - service [10](#)
- Windows service [15](#)
- write access privilege [114](#)

