IBM Cognos TM1
Version 10.2.0

Operation Guide

IBM
Note

Before using this information and the product it supports, read the information in “Notices” on page 173.

Product Information

This document applies to IBM Cognos TM1 Version 10.2.0 and may also apply to subsequent releases.

Licensed Materials - Property of IBM

© Copyright IBM Corporation 2007, 2013.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Introduction........................................... ix

Chapter 1. What's new .................................. 1
New features in Cognos TM1 Operations Guide version 10.2.0. ........................................ 1
Multi-Threaded Queries in Cognos TM1 version 10.2.0 ........................................ 1

Chapter 2. TM1 System and Feature Configuration ............ 3
Configuration Overview ................................ 3
Improving processing performance with Multi-threaded Queries ..................................... 4
Capability Assignments .................................. 5
  Cognos TM1 Application Web Capabilities .................................. 7
  Setting Capabilities ..................................... 7
  Understanding the Effect of Blanks in the Capability Assignments ................................ 8
Using Persistent Feeders .................................. 8
  Saving data with persistent feeders .................................. 9
  Modifying rules for cubes with feeders .................................. 9
  Handling corrupted or invalidated feeder files .................................. 10
Configuring and Managing Sandboxes in your TM1 Environment ................................ 10
  Memory Usage Considerations for Sandboxes and Personal Workspaces ......................... 11
  Managing Files and Folders for Sandboxes .................................. 11
  TM1 Server Configuration Parameters for Sandboxes .................................. 11
  Understanding Sandbox Differences Among Different TM1 Clients ................................ 11
Using Parallel Interaction with a Cognos TM1 server .................................. 12
  Best practice use of Parallel Interaction .................................. 13

Chapter 3. Cognos TM1 Admin Server Operations ............. 19
TM1 Admin Server ........................................ 19
Running the Cognos TM1 Admin Server on a Windows operating system ................. 20
  Running the Cognos TM1 Admin Server as an Application on Windows ......................... 21
Running the Cognos TM1 Admin Server on UNIX ........................................ 22
  Viewing Admin Server Status Report .................................. 22
Specifying the Location of the Admin Host ........................................ 22
  Specifying Multiple Admin Hosts .................................. 23

Chapter 4. Remote Cognos TM1 Server Operations ............ 25
Overview of Remote Server Operations .................................. 25
Setting Up a Remote TM1 Server to Run as an Application .................................. 26
Setting Up a Remote TM1 Server to Run as a Windows Service .................................. 27
  Installing a TM1 Server to Run as a Windows Service .................................. 27
  Removing a TM1 Server from Running as a Windows Service .................................. 27
Starting a Remote TM1 Server ........................................ 28
  Starting a TM1 Server Set Up as a Windows Application .................................. 28
  Starting a TM1 Server Installed as a Windows Service .................................. 28
  Starting a UNIX TM1 Server .................................. 29
Connecting to a Remote Server ........................................ 30
  Refreshing the List of Remote Servers .................................. 30
  Re-Setting the Admin Host .................................. 30
  Re-Setting Local Server Options .................................. 30
Disconnecting from a Remote Server ........................................ 31
Shutting Down a Windows TM1 Server ........................................ 31
  Shutting Down a Windows TM1 Server Running as a Service .................................. 31
  Shutting Down a UNIX TM1 Server .................................. 32
Shutting Down a UNIX TM1 Server Running in Background Mode .................................. 32
Logging Transactions ........................................ 33
Chapter 5. TM1 Security Overview ........................................ 43
- Overview to authentication ........................................... 43
- TM1 Object Security .................................................. 43

Chapter 6. Managing Users and Groups ............................. 45
- TM1 Users and Groups Security Overview ......................... 45
  - Implementing a TM1 Security Scheme .......................... 45
  - TM1 User and Group Security Examples ......................... 46
- Understanding Administrative Groups and Authority .......... 46
  - ADMIN Group ..................................................... 46
  - SecurityAdmin Group ........................................... 46
  - DataAdmin Group ............................................... 46
- Adding and Deleting Users and Groups ......................... 47
  - Adding a User .................................................... 47
  - Adding a Group .................................................. 47
  - Deleting a User .................................................. 48
  - Deleting a Group ............................................... 48
- Assigning Users to Groups ......................................... 49
  - Membership in Multiple Groups ................................ 49
- Securing TM1 Data .................................................. 50
  - Restricting Access to the Data Directory ....................... 50
  - Assigning Passwords ............................................ 50
  - Using Standard Security for Passwords ......................... 50
- Setting and Clearing Passwords ................................... 51
  - Setting a Password .............................................. 51
  - Clearing a Password ............................................. 51
  - Changing a Password ............................................ 51
  - Setting an Expiration for a User ................................. 52

Chapter 7. System and Performance Monitoring .................... 55
- Overview of TM1 System and Performance Monitoring ........ 55
- Using the Admin Server Log ........................................ 56
  - Message Severity Levels for Admin Server Logging .......... 56
  - Configuring Admin Server Logging ............................. 56
  - Enabling Admin Server Logging ................................ 56
  - Viewing the Admin Server Log File ............................. 56
- Logging Transactions ................................................ 57
  - Data Backup and Recovery ...................................... 57
  - Enabling and Disabling Transaction Logging .................. 57
  - Viewing the Transaction Log .................................... 57
  - Backing Out Records from the Transaction Log ............... 58
  - Removing Transaction Log Files from the Disk ................ 58
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Severity Levels for IBM Cognos TM1 Web Logging</td>
<td>122</td>
</tr>
<tr>
<td>Configuring and enabling IBM Cognos TM1 Web logging</td>
<td>123</td>
</tr>
<tr>
<td>Viewing the IBM Cognos TM1 Web Log File</td>
<td>124</td>
</tr>
<tr>
<td>Microsoft Excel .xls worksheets</td>
<td>124</td>
</tr>
<tr>
<td>Converting a .xls worksheet to .xlsx</td>
<td>125</td>
</tr>
<tr>
<td>Chapter 10. Understanding Cube Dependency</td>
<td>127</td>
</tr>
<tr>
<td>Changes in Dependency Mapping</td>
<td>127</td>
</tr>
<tr>
<td>Clearing Cube Dependencies</td>
<td>129</td>
</tr>
<tr>
<td>Creating Cube Dependencies</td>
<td>129</td>
</tr>
<tr>
<td>Using AddCubeDependency to Establish Dependencies</td>
<td>129</td>
</tr>
<tr>
<td>Debugging and Identifying Cube Dependencies</td>
<td>130</td>
</tr>
<tr>
<td>Chapter 11. Replicating Cubes</td>
<td>131</td>
</tr>
<tr>
<td>Replication Overview</td>
<td>131</td>
</tr>
<tr>
<td>Relationships Created by Replication</td>
<td>132</td>
</tr>
<tr>
<td>Required Access Privileges</td>
<td>135</td>
</tr>
<tr>
<td>Admin Server Considerations</td>
<td>135</td>
</tr>
<tr>
<td>Configuring the Tm1s.cfg File to Support Replication</td>
<td>136</td>
</tr>
<tr>
<td>Maintaining Replication Connections</td>
<td>136</td>
</tr>
<tr>
<td>Creating a Replication Connection</td>
<td>136</td>
</tr>
<tr>
<td>Modifying a Replication Connection</td>
<td>137</td>
</tr>
<tr>
<td>Deleting a Replication Connection</td>
<td>137</td>
</tr>
<tr>
<td>Replication Process</td>
<td>138</td>
</tr>
<tr>
<td>Replicating a Cube</td>
<td>138</td>
</tr>
<tr>
<td>Specifying Information about the Replicated Cube</td>
<td>139</td>
</tr>
<tr>
<td>What Happens when TM1 Replicates a Cube</td>
<td>140</td>
</tr>
<tr>
<td>Synchronization Process</td>
<td>141</td>
</tr>
<tr>
<td>Synchronizing on Demand</td>
<td>141</td>
</tr>
<tr>
<td>Scheduling Synchronization</td>
<td>141</td>
</tr>
<tr>
<td>Synchronizing over Unstable or Wide Area Network Connections</td>
<td>142</td>
</tr>
<tr>
<td>Appendix A. Accessibility features</td>
<td>143</td>
</tr>
<tr>
<td>Keyboard shortcuts</td>
<td>143</td>
</tr>
<tr>
<td>IBM and accessibility</td>
<td>144</td>
</tr>
<tr>
<td>Appendix B. Control Cubes</td>
<td>145</td>
</tr>
<tr>
<td>Security Control Cubes</td>
<td>145</td>
</tr>
<tr>
<td>CellSecurity_CubeName</td>
<td>145</td>
</tr>
<tr>
<td>ChoreSecurity</td>
<td>146</td>
</tr>
<tr>
<td>ClientSecurity</td>
<td>146</td>
</tr>
<tr>
<td>CubeSecurity</td>
<td>147</td>
</tr>
<tr>
<td>DimensionSecurity</td>
<td>147</td>
</tr>
<tr>
<td>ElementSecurity_DimensionName</td>
<td>148</td>
</tr>
<tr>
<td>ProcessSecurity</td>
<td>149</td>
</tr>
<tr>
<td>Client and Group Administration Control Cubes</td>
<td>149</td>
</tr>
<tr>
<td>ClientGroups</td>
<td>149</td>
</tr>
<tr>
<td>ClientProperties</td>
<td>150</td>
</tr>
<tr>
<td>Object Attribute and Property Control Cubes</td>
<td>151</td>
</tr>
<tr>
<td>ConnectionProperties</td>
<td>151</td>
</tr>
<tr>
<td>CubeProperties</td>
<td>151</td>
</tr>
<tr>
<td>DimensionProperties</td>
<td>152</td>
</tr>
<tr>
<td>DimensionAttributes</td>
<td>152</td>
</tr>
<tr>
<td>ElementAttributes_DimensionName</td>
<td>153</td>
</tr>
<tr>
<td>HierarchyProperties</td>
<td>153</td>
</tr>
<tr>
<td>Performance Monitoring Control Cubes</td>
<td>154</td>
</tr>
<tr>
<td>StatsByClient</td>
<td>154</td>
</tr>
<tr>
<td>StatsByCube</td>
<td>155</td>
</tr>
<tr>
<td>StatsByCubeByClient</td>
<td>156</td>
</tr>
<tr>
<td>StatsForServer</td>
<td>157</td>
</tr>
</tbody>
</table>
Introduction

This document is intended for use with IBM® Cognos® TM1®.

This document describes TM1 architecture, server operations, authentication, and component security.

TM1 integrates business planning, performance measurement and operational data to enable companies to optimize business effectiveness and customer interaction regardless of geography or structure. TM1 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 IBM Cognos product documentation on the web, including all translated documentation, access one of the [IBM Cognos Information Centers](http://pic.dhe.ibm.com/infocenter/cogic/v1r0m0/index.jsp). Release Notes are published directly to Information Centers, and include links to the latest technotes and APARs.

Samples disclaimer

The Samples Outdoors Company, GO Sales, any variation of the Great Outdoors name, 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

This product does not currently support accessibility features that help users with a physical disability such as restricted mobility or limited vision, to use this product. The TM1 Operations Console is accessible. See the Accessibility topic for more information.

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.
Chapter 1. What's new

This section contains a list of new, changed, and removed features for this release.

It will help you plan your upgrade and application deployment strategies and the training requirements for your users.

To locate the most current product documentation, go to the IBM Cognos TM1 information center (http://pic.dhe.ibm.com/infocenter/ctm1/v10r2m0/index.jsp)

New features in Cognos TM1 Operations Guide version 10.2.0

Listed below are new features in IBM Cognos TM1 Operation Guide since the last release.

**Multi-Threaded Queries in Cognos TM1 version 10.2.0**

The IBM Cognos TM1 server in version 10.2.0 can process multi-threaded queries.

Multi-threaded queries can calculate multiple streams of results in parallel to improve query performance. Significant performance improvement may be seen when processing queries if sufficient processing cores are available. Long queries can be automatically split across multiple cores increasing speed and efficiency.

For more information, see "Improving processing performance with Multi-threaded Queries" on page 4.
Chapter 2. TM1 System and Feature Configuration

This section describes how to configure the IBM Cognos TM1 server and clients.

Configuration Overview

Use the following components and features to configure the IBM Cognos TM1 server and clients.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tm1s.cfg file</td>
<td>A configuration file that specifies environment information for the TM1 server.</td>
</tr>
<tr>
<td></td>
<td>You can edit the Tm1s.cfg file to reflect the environment of the associated server.</td>
</tr>
<tr>
<td></td>
<td>For details and a complete listing of all the parameters, see the IBM Cognos TM1 Installation and Configuration Guide.</td>
</tr>
<tr>
<td>Tm1p.ini file</td>
<td>A configuration file that specifies the environment information for the TM1 clients (TM1 Perspectives, TM1 Architect, and Client).</td>
</tr>
<tr>
<td></td>
<td>For details and a complete listing of all the parameters, see the IBM Cognos TM1 Installation and Configuration Guide.</td>
</tr>
<tr>
<td>Multi-threaded Queries</td>
<td>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 executing each query on a separate core. This multiple processing can improve efficiency and processing time for large queries and rules. For more information, see &quot;Improving processing performance with Multi-threaded Queries&quot; on page 4</td>
</tr>
<tr>
<td>Capability Assignments</td>
<td>A set of capabilities that administrators can enable or disable by user group.</td>
</tr>
<tr>
<td></td>
<td>Capabilities allow you to manage options such as Personal Workspace Writeback Mode, Sandboxes, Data Reservation, and access to Server Explorer.</td>
</tr>
<tr>
<td></td>
<td>For details, see &quot;Capability Assignments&quot; on page 5.</td>
</tr>
<tr>
<td>Persistent Feeders</td>
<td>A parameter that can improve reload time of cubes with feeders, especially those with many complex feeder calculations, by saving feeders and then re-loading them at server startup without recalculating those feeders.</td>
</tr>
<tr>
<td></td>
<td>For details, see &quot;Using Persistent Feeders&quot; on page 8.</td>
</tr>
</tbody>
</table>
### Configuration Description

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
</table>
| Sandboxes         | A feature that allows users to create and work in their own data areas separate from base data. Administrator tasks for this feature include understanding the feature and its memory usage, enabling the necessary configuration parameters and managing related files and folders.  
For details, see "Configuring and Managing Sandboxes in your TM1 Environment" on page 10. |
| Parallel Interaction | A server-related feature that allows for greater concurrency of read and write operations on the same cube object. Parallel Interaction helps to keep writers to a cube from being blocked waiting for readers of the same cube to finish their read operation.  
For details, see "Using Parallel Interaction with a Cognos TM1 server" on page 12. |
| Data Reservation  | 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.  
Administrator tasks for DR include configuring related properties in the jCubeProperties control cube and the Capability Assignments window.  
To actually apply and manage Data Reservations, you must use the related TurboIntegrator and API functions that allow you to programmatically obtain, release and manage reservations.  
For complete details, see the "Using Data Reservations" and "Enabling Data Reservation" sections in the IBM Cognos TM1 Developer Guide. |

**Improving processing performance with Multi-threaded Queries**

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

Multi-threaded queries allow IBM Cognos 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.

For example:
- Query 1 executes and receives the parent thread and the 7 worker threads (8 total).
- Query 2 executes. 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 executes. The threads once again complete their sub task on the threads inflight, and then reassign to the new query. That causes a split of Query 1 into 3 threads, Query 2 – 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 assigned to Query 1 are assigned to Query 2 and Query 3 so they each maintain 4 threads.

Best practice is to set `MTQ` to the highest available combination of physical and multi-threaded cores. For example, set `MTQ=8` for 4 hyper-threaded cores or `MTQ=16` for 16 physical cores.

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 splitting the query processing threads across multiple cores.

To allow the server to process multi-threaded queries, specify the maximum number of threads to use when processing queries by adding

\[ MTQ=n \]

to the `Tm1s.cfg` where \( n \) represents the maximum overall number of threads per query.

You can track multi-threaded query execution using the Enhanced object contention report in Cognos TM1 Operation Console.

### 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 usergroup using the Capability Assignment menu found in the Server Explorer, Server Menu.

With the Server Explorer open, select a server to display the Server Menu and select **Capability Assignments**.

By default all assignments are blank. The blank settings have different implications for each capability. Changes made to these assignments take effect after you log out of TM1 then log back in. Be sure to save the settings to the server to have changes take effect.

The following capabilities can be set per usergroup:

- **Access to Server Explorer**
  
  Use this capability to launch the Server Explorer by default for this usergroup. You can set this capability to Deny to prevent the Server Explorer from being
used by this usergroup. By default, this capability is set to blank or Grant. To block access to the server explorer, click the intersection of the usergroup and the capability and select Deny.

- **Personal Workspace Writeback Mode**
  This capability defines how data changes are handled in this usergroup. 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, usergroups do not have the Personal Workspace Writeback mode capability granted. By default this capability is blank which acts like a Deny.

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

Note also that usergroups 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 7 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 usergroup 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 a Grant. To prevent this usergroup 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 IBM Cognos TM1 Installation and Configuration Guide.

• **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 IBM Cognos TM1 Developer Guide.

**Cognos TM1 Application Web Capabilities**

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

In Cognos 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 Cognos TM1 Application Web, you cannot submit from the Workflow page.

**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.

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

See the “Using Personal Workspaces and Sandboxes” in the IBM Cognos TM1 User Guide for more information about the implications of Personal Workspaces and Sandboxing on users.

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.

<table>
<thead>
<tr>
<th>Writeback</th>
<th>Action</th>
<th>Sends Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>Read--&gt; Recalc</td>
<td>From the Base TM1 Model.</td>
</tr>
<tr>
<td></td>
<td>Write--&gt; Recalc</td>
<td>To the Base TM1 Model.</td>
</tr>
<tr>
<td>Personal Workspace</td>
<td>Read--&gt; Recalc</td>
<td>From the Base TM1 Model.</td>
</tr>
<tr>
<td></td>
<td>Write--&gt; Recalc</td>
<td>To the Personal Workspace.</td>
</tr>
<tr>
<td>Personal Workspace</td>
<td>Commit</td>
<td>Sends the Personal Workspace data to the TM1 Base Model.</td>
</tr>
</tbody>
</table>
| Personal Workspace with Queuing | Commit        | Sends the Personal Workspace to the Queue
|                           |                 | When all resources are available, the Personal Workspace data is processed and sent to the Base TM1 Model. |

Using 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 probably improve their performance using this parameter.

For more details about the associated parameter for this feature called CubeSaveData, see the IBM Cognos TM1 Installation and Configuration Guide.
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.

**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 Reference Guide for more information.

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 TM1 .Server All persistent feeder files will be deleted and server will shut down


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.

Configuring and Managing Sandboxes in your TM1 Environment

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.

For more details on using the Sandbox or Personal Workspace feature, see the information on Sandboxes and Personal Workspaces in the IBM Cognos TM1 User Guide and the IBM Cognos TM1 TurboIntegrator Guide.
Memory Usage Considerations 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`. For more details, see the list of parameters in the *IBM Cognos TM1 Installation and Configuration Guide*.

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 *IBM Cognos TM1 Installation and Configuration Guide* 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.

- `DisableSandboxing` parameter
- `MaxUserSandboxSize` parameter

For more details about the `Tm1s.cfg` file and these parameters, see the *IBM Cognos TM1 Installation and Configuration Guide*.

- Capability Assignments

  The use of Personal Workspaces or Sandboxes is controlled using the Capability Assignments option available on the Server Explorer, Server menu. See the System Configuration chapter of the *IBM Cognos TM1 Operation Guide* for more information.

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.

---

**Using Parallel Interaction with a Cognos TM1 server**

Parallel Interaction is an IBM Cognos TM1 server-related feature that allows for greater concurrency of read and write operations on the same cube object.

Using Parallel Interaction, writers to a cube will not be blocked waiting for readers of the same cube to finish their read operation. For example, **without** the Parallel Interaction feature, a user that is writing data to a cube can be blocked and forced to wait if another user is performing an intensive read operation on that same cube, such as recalculating a large complex view.

**Note:** As of Cognos TM1 version 10.1, Parallel Interaction is enabled by default.

**When to Use Parallel Interaction**

Parallel Interaction is most useful for cubes that typically experience a large combination of both read and write activity.

In this case, Parallel Interaction provides an improvement over the Cognos TM1 legacy concurrency mode by allowing read and write operations to execute at essentially the same time. In some cases, read operations might take a little longer when using Parallel Interaction.

**Memory usage considerations**

Cognos TM1 might possibly use more memory when Parallel Interaction is enabled as compared to using Cognos TM1 in legacy mode without this feature.

Exact memory usage varies depending on the specific cubes in your Cognos TM1 server and the amount of read/write activity on those cubes.

**Enabling Parallel Interaction for the Cognos TM1 server**

Parallel Interaction is enabled by default.

You can control this feature using the global `ParallelInteraction` configuration parameter in the `Tm1s.cfg` file to enable or disable the feature for all of the cubes in a Cognos TM1 server.

For more details, see the list of configuration parameters in the *IBM Cognos TM1 Installation and Configuration Guide*. 
Best practice use of Parallel Interaction

To obtain maximum performance gains provided by Parallel Interaction, follow these best practice guidelines.

Parallel Interaction (PI) is a server configuration that improves response time of writing data by removing lock contention resulting from concurrent cube data access (when the system reads data or when the system writes data). When enabled, Parallel Interaction changes the Cognos TM1 Object Locking model so that write operations are not blocked by concurrent data read or data write operations to the same cube (or dependent cubes based on rules). Read operations always includes the most current write activity as of the time the read operation begins.

Comparisons between Cognos TM1 with or without Parallel Interaction show significant performance improvements for writer response time and an overall greater throughput of transactions with test cases that simulate high user concurrency.

Cognos TM1 Parallel Interaction provides immediate writer response times in cases where writer performance had been negatively impacted without Parallel Interaction. Improvements include:

- Support of concurrent writers executing high-level data spreading input
- Support of high user counts for readers demanding instantaneous access to freshly-entered data
- Maintenance of more recently-updated environments during cube data loads over the course of a day
- Execution of TurboIntegrator Data Allocation processes sourcing data from Input Cubes
- Execution of TurboIntegrator Data Manipulation processes to incrementally build Reader Cubes sourcing data from Input Cubes

Parallel Interaction is useful for any Cognos TM1 application involving writeback activity. Sufficient server CPU cores are needed to realize the benefit of Parallel Interaction and to process the greater throughput of concurrent read and write operations. Sufficient cores also ensure that any remaining “blocking” activity resulting from meta data maintenance does not interfere with writer performance. Parallel Interaction does not improve query performance, so it is not recommended for Cognos TM1 read-only applications.

Parallel Interaction impact on Cognos TM1 server core utilization

If the environment has sufficient server cores to process the incremental transactional activity, using Parallel Interaction in Cognos TM1 for work that is subject to reader/writer contention generates higher transaction throughput than similar work performed in previous versions of Cognos TM1. With Parallel Interaction writers are no longer blocked by concurrent readers or other writers to the same cube (and dependent cubes); therefore, the Cognos TM1 server cores tend to process more transactions during periods of high concurrency.

Customers observe significantly higher CPU utilization using Cognos TM1 Parallel Interaction during periods of high read/write concurrency as compared to previous versions of Cognos TM1.
Insufficient Cognos TM1 server core capacity may negatively impact the potential performance advantages of Parallel Interaction. Cognos TM1 customers upgrading to Cognos TM1 9.5.2 or later from prior versions should monitor Cognos TM1 server CPU utilization when operating with Parallel Interaction to ensure that server cores are not "saturated," which is indicated by a greater than 80% server utilization for moderate to long periods of time. If use of Parallel Interaction results in Cognos TM1 Server core saturation, add additional cores to the server as available or consider a server with more available cores.

**Parallel Interaction impact on reader performance**

In scenarios involving high volume mixed read/write activity, the read operations may be slightly slower in Cognos TM1 9.5.2 or higher with Parallel Interaction On compared to Cognos TM1 9.5.1. This situation occurs because with Parallel Interaction, writers utilize CPU resources more efficiently and compete with readers for those resources. With the greater throughput of write activity in Cognos TM1 9.5.2 or higher, Parallel Interaction more frequently invalidates internal caches which can also negatively impact reader performance.

Parallel Interaction does not yield performance benefits for readers in read-only Cognos TM1 applications.

**Parallel Interaction's impact on Cognos TM1 Semaphore best practice**

Prior to Cognos TM1 9.5.2, customers employed a TurboIntegrator Best Practice called "Semaphore" to prevent TurboIntegrator process "thrashing" (a wasteful situation of successive rollbacks and retries which results in performance issues for the end users). Thrashing is caused by concurrent execution of TurboIntegrator processes that locked the same objects and blocked each other.

For instance, consider an end user data input Websheet with an Action Button that calls a process to copy data from input cubes to a reporting cube. This operation could result in concurrent execution of a process that performs a read operation followed by a write operation. During periods of high concurrency, the write operation could encounter a locked cube (as a result of another write operation), causing the process to rollback its work and re-initiate the process. Subsequent re-tries by the process could also encounter a locked destination cube, causing another rollback and resulting in a thrashing scenario that negatively impacts end-user performance.

The Cognos TM1 Semaphore Best Practice approach required the TurboIntegrator processes to update a common cube in the Prolog section to register themselves for execution; this action meant that if the process held the lock on the semaphore cube, it would execute to completion without interference from locks by concurrently-run processes. If the attempt to write to the common cube in the Prolog encountered a lock, the rollback time had, by comparison, a minimal impact on performance. With Parallel Interaction, concurrent writes to the semaphore cube do not cause serialization; all writes occur in parallel.

With Cognos TM1 9.5.2 Parallel Interaction, the Cognos TM1 Semaphore Best Practice as described above may not be needed or may need to be adjusted to ensure a similar behavior. If the process is only reading and writing data, the Semaphore approach is no longer needed to achieve better performance. In this case the concurrent processes run to completion with no locking behavior.
If the process is performing meta data maintenance, a new TurboIntegrator function called synchronized() is available to ensure that the process doesn’t cause unnecessary usage or contention of Cognos TM1 and system resources. For more details, see the topic “Serializing TurboIntegrator processes using synchronized()” in the IBM Cognos TM1 TurboIntegrator Guide.

Performing maintenance to a common dimension might be a suitable alternative. The Best Practice for using a dimension as a semaphore controlling object requires that the Base TurboIntegrator process calls a Dimension Semaphore process using Execute Process in the Prolog tab. The Dimension Semaphore process will simply insert a dummy element in the controlling dimension, thereby locking that dimension so no other process that maintains that dimension can proceed. The Base process then removes the dummy element in the Epilog tab.

**Parallel Interaction impact on use of Batch Update Start and Batch Update Finish Wait**

The existing Best Practice to improve data loading performance executes multiple TurboIntegrator processes in parallel, wrapping each of them with Batch Update Start/Batch Update Finish Wait functions. This approach is commonly used to split up a large data load into several smaller data loads (the count depends on the cores available to manage the data load). Executing processes within Batch Update functions allows each process to operate independently up to the point of taking an IX lock on the target cubes. At that time the first finishing process updates the cube and the others follow sequentially. The end result is a faster data load than a single TurboIntegrator process.

Parallel Interaction allows for simultaneous execution of processes reading and/or writing to the same cube (none of the data loading processes blocking the others) without the need for Batch Update Start / Batch Update Wait Finish. Remember that processes doing meta data updates result in blocking actions.

Cognos TM1 9.5.2 tests with Parallel Interaction On compared with simultaneous data loads using Batch Update reveal that performance is faster without Batch Update. Therefore, when Parallel Interaction is On, it is best to remove the use of Batch Update from TurboIntegrator processes.

For read-only environments where Parallel Interaction is Off, Batch Update Start/Batch Update Finish Wait remains a Best Practice for increasing the performance data loading using simultaneous data loading TurboIntegrator processes.

**Parallel Interaction impact on Cognos TM1 server memory**

Internal testing comparing Cognos TM1 9.5.1 to Cognos TM1 9.5.2 with Parallel Interaction have shown the Cognos TM1 9.5.2 Server to allocate between 10% to 30% more RAM than the Cognos TM1 9.5.1 server. Some of this increased memory use is due to the internal structures created by Parallel Interaction, however the additional throughput of transactions and their impact on View Cache is also a contributing factor. Plan for 30% more memory usage with Cognos TM1 9.5.2 than with prior versions of Cognos TM1.
Operations that cause object locking and potential performance delays

Parallel Interaction greatly improves writer performance and provides predictable performance as reader and writer user counts scale. However, there are Cognos TM1 operations that can mask the benefits of Parallel Interaction by locking objects resulting in response-time performance delays.

- Establishing Cube Dependencies
  To ensure proper data integrity and cache management in cross-cube references in rules (DB statements or ATTRN/S statements), internal relationships (dependency) between cubes must be created. Establishing the cube dependency is a quick operation, but when it is performed as part of a long running view or TurboIntegrator process, it blocks concurrent read and write activity to the dependent cube (Cube A feeds Cube B, so Cube B is "dependent" on Cube A, and the dependency property setting on Cube B blocks activity to Cube B). Once a Cube Dependency is established, it remains valid for the duration of the server session unless it becomes invalidated by a Rule file or dimension maintenance.

  During Server load, Cognos TM1 establishes cube dependencies based on feeder statements in rule files. Feeder statements with data-dependent cube references or rules files containing ATTRN or ATTRS functions do not establish Cube Dependencies during server load; rather those dependencies are established during a first-time query or data update invoking these cross-cube rules.

  To improve this situation, create a TurboIntegrator process that establishes all potential cube dependencies in the model using the AddCubeDependency function for each cube-dependent relationship established by rules. The process containing these explicit Cube Dependency functions should be executed immediately following a Cognos TM1 Server Startup, after rule file changes, and following dimension maintenance. This activity quickly establishes server-wide cube dependencies and eliminates the possibility that a long running query or process will trigger a cube dependency during periods of user activity that might block objects and cause contention issues for concurrent readers and writers.

  Cube dependencies can be determined by scanning rules files for DB statements and ATTRN and ATTRS functions. An ATTRN/S function creates a dependency on the base cube's corresponding element attribute cube. Alternatively, a system administrator can add the following statement to their Tm1s-log.properties file to have Cognos TM1 log each time it creates a cube dependency:

  Log4j.logger.TM1.Cube.Dependency=INFO

  Entries in the log file will look like the following:


  Adding cube dependency: cube "SalesCube" depends on cube "PriceCube"

  In this example, PriceCube is the base cube and SalesCube is the dependent cube. The correct application of this dependency when creating an explicit AddCubeDependency function is as follows:

  AddCubeDependency('PriceCube','SalesCube')

  See Chapter 10, “Understanding Cube Dependency,” on page 127 for more information.

- ViewConstruct in a TurboIntegrator Process

  Use a ViewConstruct function in TurboIntegrator processes to pre-cache a view to improve performance for subsequent View execution (assuming the cache has not been invalidated). This function blocks reads and writes to the View's cube for the duration of the process.

  See the ViewConstruct function in the IBM Cognos TM1 Reference Guide.
- **Element Maintenance**
  Element Maintenance will cause blocks to readers and writers of any cube containing the maintained dimension. Contention issues may arise when the dimension is maintained as part of a longer running TurboIntegrator process. Isolate meta data maintenance from data maintenance in distinct TurboIntegrator Processes so that the meta data maintenance operation doesn't carry its lock throughout the data maintenance operation (which is non-blocking with Parallel Interaction). This goal can be accomplished in separately-executed Chores or by using the TM1RunTI utility to execute processes external to Cognos TM1. The TM1RunTI utility was provided in TM1 9.5.2 minor follow-up release and is part of the standard installation as of Cognos TM1 10.1.0.

- **Updating an Alias**
  Updating an alias attribute causes blocks to readers and writers of any cube containing the maintained dimension. Contention issues arise when the dimension is maintained as part of a longer running TurboIntegrator process. Isolate alias maintenance in TurboIntegrator Processes to lessen the blocking impact.
Chapter 3. Cognos TM1 Admin Server Operations

This section provides an overview of the IBM Cognos TM1 Admin Server and describes the typical administrator tasks of running the Admin Server on Windows and UNIX systems.

For details about Cognos TM1 system architecture, see the IBM Cognos TM1 Installation and Configuration Guide.

TM1 Admin Server

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 Cognos TM1 Admin Server on a Windows operating system

After you install the IBM Cognos 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 *IBM Cognos TM1 TurboIntegrator Guide* for more information.

**Manually starting the Cognos TM1 Admin Server on Windows**

You can manually run the Cognos 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 Cognos 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 Cognos 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 an Windows application. For details, see "Running the Cognos TM1 Admin Server as an Application on Windows" on page 21.
Using Cognos Configuration to manage the Cognos TM1 Admin Server on Windows

For details on using the Cognos Configuration utility to manage the Cognos TM1 Admin Server on a Microsoft Windows operating system, see the IBM Cognos TM1 Installation and Configuration Guide.

Running the Cognos 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 Cognos TM1. 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. That way, 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 Cognos 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 Cognos 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 Cognos 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 Cognos 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 the *IBM Cognos TM1 Installation and Configuration Guide*.

**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.
- 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 the *IBM Cognos TM1 Installation and Configuration Guide*.
- 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 the *IBM Cognos TM1 Installation and Configuration Guide*. 

---

IBM Cognos TM1 Version 10.2.0: Operation Guide
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 4. Remote Cognos TM1 Server Operations

This section describes how IBM Cognos TM1 remote servers work, and how you can manage their operations.

For complete details about Cognos TM1 system architecture, see the IBM Cognos TM1 Installation and Configuration Guide.

For details on using the IBM Cognos Configuration utility to manage TM1 remote servers on a Windows or UNIX system, see the IBM Cognos TM1 Installation and Configuration Guide.

Overview of Remote Server Operations

The following figure illustrates the operations of a remote TM1 server. These operations are explained in the text that follows.

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.

- On startup, the remote server loads dimensions and cubes from the data directory into the server machine RAM. At the same time, the server opens a new transactional log file called Tm1s.log in the data directory. After the cubes are loaded, the remote server is available.
- The remote TM1 server registers itself with one or more Admin Servers so that clients can connect to the remote TM1 server.
• Client applications contact Admin Servers to locate available TM1 servers. The clients log into the TM1 servers whose data they want to access.

• Clients edit the cube data, sending the values back to the TM1 server.

• As new values are received from clients, the TM1 server writes the records to the Tm1s.log file, keeping track of every data change, including the date and time the edit occurred, and the ID of the client who made the edit.

• As the server calculates new values in response to client requests, the server stores them in memory, increasing the amount of memory used by the server.

• When the server shuts down, all records in the Tm1s.log file are saved to disk, and the transaction log file is renamed by appending a date/time stamp to it. The Tm1s.log file is saved in the server's data directory to back out data transactions, as described in “Back Out Records from the Transaction Log” on page 36.

If the server is intentionally shut down without saving the changes, the log file is saved with a time/date stamp and the extension is changed to .rej. You can process the Tm1syyyyymmddhhmmss.rej file through TurboIntegrator to recover the transactions.

• To save all changes to the data on a TM1 server at any time without shutting down the server, right-click a server in Server Explorer and Click Save Data. All records in the Tm1s.log file are immediately written to disk, the transaction log file is renamed by appending a date/time stamp to it, and a new Tm1s.log file is created to accept any subsequent edits to cube values.

Setting Up a Remote TM1 Server to Run as an Application

After you run the TM1 Installation Wizard on a Microsoft Windows system and install the sample TM1 servers, you can use the IBM Cognos Configuration utility to run each sample server as a Microsoft Windows service. As an alternative, you can manually set up the TM1 server to run as an application.

Procedure
1. Open the install_dir\bin directory in Windows Internet Explorer.
2. Right-click Tm1s.exe.
3. Select Create Shortcut.
   Windows creates Shortcut to Tm1s.exe in the install_dir\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:
8. Click OK to save the shortcut. You might want to move the shortcut to the desktop for easy access.

**Setting Up a Remote 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 `install\bin` directory.

The topics in this section describe how to configure a TM1 server to run as a Windows service.

### Installing a TM1 Server to Run as a Windows Service

To install a TM1 server to run as a Windows service, use the following command line format:

```
rm1sd.exe -install parameters
```

The following table lists the available parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>Required</td>
<td>Name of the TM1 Server. This will become the Windows service name as shown when you display the properties of an installed service.</td>
</tr>
<tr>
<td>-z</td>
<td>Required</td>
<td>Data directory that contains the tm1.cfg file for the TM1 database.</td>
</tr>
<tr>
<td>-u</td>
<td>Optional</td>
<td>Valid user name for the computer on which you are working. Use one of the following formats: Domain and user name in the format <code>Domain\username</code>. 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.</td>
</tr>
<tr>
<td>-w</td>
<td>Optional</td>
<td>Password for the above account.</td>
</tr>
</tbody>
</table>

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
```

### 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.
Starting a Remote TM1 Server

The topics in this section describe how to start all versions of the TM1 server.

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.

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.

tmplsd.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.

tmplsd.exe -remove -n sdata
3. Select Automatic as the Startup Type and click OK.

**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 install_dir/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 install_dir/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 install_dir/bin.

   **Note:** You must set both tm1_path and TM1_PATH to the same directory.
5. Save the script.
   For example:
   ```bash
   > ./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.

**Example**

Note that if you use an absolute path to start the server, you must use an absolute path to stop it. Similarly, if you use a relative path to start the server, you must use a relative path to stop it. For example, if you use:
   ```bash
   > ./startup_tm1s.sh "../samples/tm1/SData"
   ```
   then the shutdown script must also use the same relative path:
   ```bash
   > ./shutdown_tm1s.sh "../samples/tm1/SData"
   ```

You cannot mix and match the path description methods between startup and shutdown of the same database.

**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 the IBM Cognos TM1 Installation and Configuration Guide.
When you run the TM1 server in background mode, you must use `shutdown_tm1s.sh` to shut down the server. See "Shutting Down a UNIX TM1 Server Running in Background Mode" on page 32.

## 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.

## Re-Setting 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.

## Re-Setting Local Server Options

The following table describes the Local Server options that you can change through the TM1 Options dialog box.
<table>
<thead>
<tr>
<th>Local Server Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Directory</td>
<td>Directory for the local server data files.</td>
</tr>
<tr>
<td>Connect to Local Server on Startup</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Disconnecting from a Remote Server**

You can disconnect from a remote server.

**Procedure**

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

**Shutting Down a Windows TM1 Server**

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.

**Shutting Down a Windows TM1 Server Running as a 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 37](#).

**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.

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 shutdown_tm1s.sh located in the install_dir/bin directory.

The shutdown_tm1s.sh file accepts the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>The name of the TM1 server you want to shut down. Note that this server must have been started in background mode.</td>
</tr>
<tr>
<td>-v</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>IP address of the Admin Server on which the TM1 server is registered.</td>
</tr>
<tr>
<td>-user</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>User name of the person bringing down the server. This person must have the necessary access and privileges to shut down the server.</td>
</tr>
<tr>
<td>-pwd</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>Password for the person bringing down the server.</td>
</tr>
<tr>
<td>-time</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Delay, in seconds, at which the server will be shut down after running shutdown_tm1s.sh</td>
</tr>
<tr>
<td>-cancel</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Cancels any previously issued shutdown_tm1s.sh commands. Valid values are T and F.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>-unsave</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>By default, TM1 saves the data before a TM1 server is shut down with the <code>shutdown_tm1s.sh</code> command. The -unsave parameter shuts down a server without saving the data. Valid values are T and F.</td>
</tr>
</tbody>
</table>

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.

```
./startup_tm1s.sh -n TM1SERV -v 207.110.100.206 -user ADMIN -pwd Administrator -time 30 -unsave T
```

Note that if you use an absolute path to start the server, you must use an absolute path to stop it. Similarly, if you use a relative path to start the server, you must use a relative path to stop it. For example, if you use:

```
> ./startup_tm1s.sh "../samples/tm1/SData"
```

then the shutdown script must also use the same relative path:
```
> ./shutdown_tm1s.sh "../samples/tm1/SData"
```

You cannot mix and match the path description methods between startup and shutdown of the same database.

### 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.

The information recorded in the log file includes:
- 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.
- Optional user-provided string that is attached to each transaction. You create the string with the `TM1ServerLogSetFlagString` API function.

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","1998060602212744","19980602212744","Admin","N","430.","600.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Sep","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Oct","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Nov","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Dec","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Jan","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Feb","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Mar","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Apr","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","May","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Jun","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Jul","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Aug","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Sep","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Oct","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Nov","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Dec","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Jan","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Feb","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Mar","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Apr","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","May","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Jun","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Jul","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Aug","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Sep","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Oct","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Nov","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Dec","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Sedan","Units","Jan","199806062212749","199806062212749","Admin","N","610.","800.","salescube","Budget","Belgium"
Data Backup and Recovery

The Tm1s.log file remains open while the TM1 server is running.

When you bring down the server normally, TM1 renames the log file by appending a time stamp to it, and with the following naming convention:

Tm1syyyymmddhhmmss.log

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 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.

Enabling and Disabling 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.

You can enable logging for individual cubes.

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.

**Note:** You must use a 9.5 or newer TM1 client to view log files on a 9.5.x server. The transaction log does not display correctly when you use a pre-9.5 client to view log files on a 9.5.x server.

**Procedure**

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting Time</td>
<td>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.</td>
</tr>
<tr>
<td>End Time</td>
<td>End date and time for the query. The date and time format is MM/DD/YYYY HH:MM:SS. The default is <strong>/</strong>/____ <strong>:</strong>:__, 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.</td>
</tr>
<tr>
<td>Clients</td>
<td>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 (*).</td>
</tr>
<tr>
<td>Cubes</td>
<td>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 (*).</td>
</tr>
</tbody>
</table>

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. At any time after you start the query, you can click Cancel to stop the search. When you choose Cancel, nothing is returned for your search.

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 the Tm1s.cfg file topic in the IBM Cognos TM1 Installation and Configuration Guide.

Backing Out Records from 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 the record(s) you want to back out.
   - To highlight an individual record, click the record.
   - To highlight multiple adjacent records, click the first record and SHIFT+click the last record.
   - To highlight multiple non-adjacent records, CTRL+click each record.
2. Click Edit, Select.
   - All the highlighted records now display a check mark in the box adjacent to the first column. The check marks indicate that the record is selected to be backed out.
   - To select all the records without first highlighting them, click Edit, Select All.
3. Click Edit, Back Out.
   - TM1 backs out the records in reverse chronological order as identified by the LOGTIME column.

Removing 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.

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.

The following control cubes are populated during performance monitoring. For details, see Appendix B, “Control Cubes,” on page 145.
### Cube Description

<table>
<thead>
<tr>
<th>Cube</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StatsByClient</td>
<td>Tracks the message count, average message size, total elapsed time, and other measures for each client on the server.</td>
</tr>
<tr>
<td>StatsByCube</td>
<td>Tracks the memory used for each cube on the server.</td>
</tr>
<tr>
<td>StatsByCubeByClient</td>
<td>Tracks the number and elapsed time of cell updates, cell retrievals, view calculations, and view retrievals for each client and cube on the server.</td>
</tr>
<tr>
<td>StatsForServer</td>
<td>Tracks the connected clients, active threads, and memory used by the server.</td>
</tr>
</tbody>
</table>

### Procedure

1. Select the server in Server Explorer.
2. Click **Server, Start Performance Monitor**.
   - 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.
   - For more details about configuration parameters, see the *IBM Cognos TM1 Installation and Configuration Guide*.
   - To end performance monitoring during a server session:
3. Select the server in Server Explorer.
4. Click **Server, Stop Performance Monitor**.

### 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 IMStatsForServer control cube, which is described in Appendix B, “Control Cubes,” on page 145.

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 the IBM Cognos TM1 Installation and Configuration Guide.

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

Some examples of using Bulk Load Mode include:

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

Cognos TM1 typically runs in a multi-user mode where multiple users, chores and processes can all run concurrently accessing Cognos TM1 data. In Bulk Load Mode, the Cognos 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 Cognos TM1. As soon as Bulk Load Mode is done, any users that were previously logged in are reactivated and user-interaction with Cognos 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 Cognos 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 and TM1Top 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(TMIP hPool, TMIServer hServer)
TM1V TM1ServerDisableBulkLoadMode(TMIP hPool, TMIServer hServer)

where:

• TMIP is a pool handle obtained with TM1ValPoolCreate.
• TMIServer 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 TI
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 TI 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 TI 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 following TI commands to enable and disable Bulk Load Mode in a TI process.

EnableBulkLoadMode()

DisableBulkLoadMode() - This function can only be used on the last line in the Epilog section of your TI process when using Bulk Load Mode.
Chapter 5. TM1 Security Overview

You can control access and authentication to the servers.

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

For complete details about configuring authentication security, see the *IBM Cognos TM1 Installation and Configuration Guide*.

Overview to authentication

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

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

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

Table 1. Cognos TM1 server authentication methods

<table>
<thead>
<tr>
<th>Authentication Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM1 Authentication</td>
<td>Cognos TM1 server asks you for a user name and password, and validates the login information against the security cube login information.</td>
</tr>
<tr>
<td>Integrated Login</td>
<td>Microsoft Windows performs the Cognos TM1 authentication.</td>
</tr>
<tr>
<td>LDAP Authentication</td>
<td>Cognos TM1 server asks you for a user name and password, and validates the login information against an external LDAP server.</td>
</tr>
<tr>
<td>IBM Cognos Security</td>
<td>The Cognos TM1 server authenticates users using IBM Cognos security.</td>
</tr>
</tbody>
</table>

TM1 Object Security

As the TM1 administrator, you can control access to TM1 objects by assigning specific levels of object security to TM1 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.

For more information about the procedures required to set security for TM1 objects, see the *IBM Cognos TM1 Developer Guide*.
Chapter 6. Managing Users and Groups

This section explains how to manage users and groups in IBM Cognos TM1. In TM1, security is based on the groups users belong to.

TM1 Users and Groups Security Overview

TM1 manages security by organizing TM1 users into groups. TM1 includes a set of three predefined administrative groups and also allows you to 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 and SecurityAdmin. For more details, see "Understanding Administrative Groups and Authority".
- **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 on your machine to administer security on the IBM Cognos TM1 server.

Implementing a TM1 Security Scheme

You can implement a TM1 security scheme to secure objects on a remote server, in the following way:

**Procedure**

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

The security levels you can implement are:

- **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 using a TM1 username and password. This security is completely independent of standard file system security and is not affected by user rights on network directories or file servers.
TM1 User and Group Security Examples

To illustrate IBM Cognos TM1 security and allow you to experiment with security features, the TM1 sample data (Sdata) that is installed with the IBM Cognos TM1 server 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.

<table>
<thead>
<tr>
<th>User Name</th>
<th>Password</th>
<th>Group Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin</td>
<td>apple</td>
<td>ADMIN</td>
</tr>
<tr>
<td>Usr1</td>
<td>red</td>
<td>North America</td>
</tr>
<tr>
<td>Usr2</td>
<td>orange</td>
<td>South America</td>
</tr>
<tr>
<td>Usr3</td>
<td>blue</td>
<td>North America and South America</td>
</tr>
</tbody>
</table>

Understanding Administrative Groups and Authority

TM1 supports the separation of administrative duties and roles in TM1 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.

- **DataAdmin group** - Members in the DataAdmin group have 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.

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

The security assignments for these three administrator groups are hard-coded and can not 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 only be performed by members of the ADMIN group. Members of the DataAdmin and SecurityAdmin groups do not have all the required access privileges to perform these operations.

The following sections provide details about each administrative group.

**ADMIN Group**

The ADMIN group, which is created for administrative purposes only, should contain only a very 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:
- Assign and revoke other user’s access rights.
- 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.

Joint membership in the ADMIN group and SecurityAdmin group or the ADMIN group and the DataAdmin group is treated as ADMIN.

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

**SecurityAdmin Group**

The SecurityAdmin group can only perform security operations in TM1. Members can access the security related features of TM1, but can not 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 rights 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 using 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 can not write or modify rules and processes. They can not view rules, but can view processes in read-only mode.

For more details about processes, see the IBM Cognos TM1 Developers Guide.

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 is 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 can not 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 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 rights to a cube, the user will still be able to see the cube based on their DataAdmin rights.
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 rights 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 can not create rules on security control cubes. This restriction prevents a DataAdmin from creating a rule that could modify cells in a security control cube.

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

The IBM Cognos TM1 server can contain up to 65,535 groups. The GroupsCreationLimit parameter in the server’s TM1s.cfg file determines the number of groups you can create during a single TM1 server session.
**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.

   **Note:** By default, you can add up to 20 groups per session. To increase the number of groups you can add per session, change the value of the GroupsCreationLimit parameter in the TM1s.cfg file. For more details about configuration parameters, see the *IBM Cognos TM1 Installation and Configuration Guide*.

**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 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 a group.

**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 rights from all groups.

For example, in the sample data, Usr3 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 Usr3 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 TM1 for high-level planning and analysis, TM1 data can be sensitive and confidential.

TM1 provides all the tools you need to secure your data, but as with any security system, real security is only as good as the procedures you implement. We suggest implementing the following procedures to enhance TM1 security:

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 rights similar to those of super users or root users on network systems; they can go anywhere and do anything.

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

Using Standard Security for Passwords

Treat TM1 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.
   TM1 asks you to 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 re-establish 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 7. System and Performance Monitoring

There are log files, tools, and utilities available for monitoring the system performance of an IBM Cognos TM1 server.

Overview of TM1 System and Performance Monitoring

TM1 includes a collection of tools that provide logging and realtime performance monitoring of the TM1 servers in your organization.

You can monitor TM1 performance and activity using the following tools.

- **Message Logging**
  - **Admin Server log**
    A log file containing messages about communication between TM1 clients, the Admin Server and individual TM1 servers.
  - **Transactions Log**
    A log file that tracks the data transactions made by the clients logged into 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 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.

- **Monitoring Server Performance Using Control Cubes**
  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.

- **TM1 Top**
  A utility that dynamically monitors the threads and processes running in an instance of a single TM1 server. TM1 Top is a stand-alone utility that runs within a console (command) window on a Microsoft Windows computer.

- **Performance Counters**
  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.

- **TM1 Operations Console**
  A web-based utility that displays and logs TM1 server activity. See the *IBM Cognos TM1 Operations Console Guide* for more information.
Using the Admin Server Log

The TM1 Admin Server log is useful for troubleshooting connection issues when using the TM1 Secure Socket Layer (SSL) with custom certificates or certificates from the Microsoft Windows certificate store.

It contains messages about the communication between TM1 clients, the TM1 Admin Server, and individual TM1 servers.

Most of the messages in this 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 the "Authentication and security configuration" chapter in the IBM Cognos TM1 Installation and Configuration Guide.

Message Severity Levels for Admin Server Logging

The Admin Server log categorizes messages into three severity levels.

These levels are also used in the logging properties file to configure logging to a specific level.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUG</td>
<td>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.</td>
</tr>
<tr>
<td>INFO</td>
<td>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.</td>
</tr>
<tr>
<td>ERROR</td>
<td>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.</td>
</tr>
</tbody>
</table>

Configuring Admin Server Logging

Configuration for TM1 Admin Server logging is contained in the tm1admsrv-log.properties file. This file is located in the install_dir\bin directory.

This file is located in the install_dir\bin directory.

The sections of the properties file are:

- **Logger section**
  
  Configures the TM1 sub-components 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 logging
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 every-day use and does
not typically require adjustment. Contact customer support for assistance if you
need to configure the logging properties file for troubleshooting purposes.

**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 install_dir\bin directory.

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

**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 \install_dir\bin 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:

<table>
<thead>
<tr>
<th>Thread ID</th>
<th>Message Level</th>
<th>Date and Time</th>
<th>Subcomponent Name</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>3328</td>
<td>DEBUG</td>
<td>2007-01-19 19:14:04, 396</td>
<td>TM1.Event</td>
<td>Timwed wait for event 0043B858 succeeded</td>
</tr>
<tr>
<td>3328</td>
<td>DEBUG</td>
<td>2007-01-19 19:14:04, 396</td>
<td>TM1.Event</td>
<td>Destroy event 0043B858 succeeded</td>
</tr>
</tbody>
</table>

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.

The information recorded in the log file includes:

- 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.
- Optional user-provided string that is attached to each transaction. You create the string with the TM1ServerLogSetFlagString API function.

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",
```
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: Tm1syyyyymmddhhmmss.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.

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 **Server, View 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting Time</td>
<td>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.</td>
</tr>
<tr>
<td>End Time</td>
<td>End date and time for the query. The date and time format is MM/DD/YYYY HH:MM:SS. The default is <strong>/</strong>/____ <strong>:</strong>:__, 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.</td>
</tr>
<tr>
<td>Clients</td>
<td>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 (*).</td>
</tr>
<tr>
<td>Cubes</td>
<td>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 (*)</td>
</tr>
</tbody>
</table>

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.

**Back out Records from 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 the record(s) you want to back out.
   - To highlight an individual record, click the record.
   - To highlight multiple adjacent records, click the first record and SHIFT+click the last record.
   - To highlight multiple nonadjacent records, CTRL+click each record.

2. Click Edit, Select.
   All the highlighted records now display a check mark in the box adjacent to the first column. The check marks indicate that the record is selected to be backed out.
   To select all the records without first highlighting them, click Edit, Select All.

3. Click Edit, Back Out.
   TM1 backs out the records 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 Corrupt 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 corrupt. If this happens, the server will not be able to restart.

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

Determining if the Transaction Log File is Corrupt

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 corrupt, the server will not start and informs you of the situation via visual prompts and/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.

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 trying 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 corrupt. An example messages in the 
TM1 Server Message log indicating a corrupt transaction file and the warning 
messages are shown in the following table.

<table>
<thead>
<tr>
<th>Thread ID</th>
<th>Message Level</th>
<th>Date and Time</th>
<th>Subcomponent Name</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>4076</td>
<td>FATAL</td>
<td>2007-10-30 18:20:10, 203</td>
<td>TM1 .Server</td>
<td>Bad transaction log record: 828.7...</td>
</tr>
</tbody>
</table>

**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 will continue running, but the problem that caused the 
unexpected shutdown may still exist.

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

Select **OK** to shutdown the TM1 server.

**Resolving a Corrupt 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. At this point, you 
should be able to successfully restart the server, but you should also contact 
Customer Support for assistance in resolving the cause of the unexpected 
shutdown.

**Using the TM1 Server Message Log**

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.

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 Cognos 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.

**Message Severity Levels**

The logging system categorizes messages into severity levels.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUG</td>
<td>Detailed, technical messages that are useful when customer support or engineering need to debug the application. Examples: &quot;SSL Connection opened&quot;, &quot;Removing SQL Connection from memory&quot;</td>
</tr>
<tr>
<td>INFO</td>
<td>Informational messages that highlight the progress of the application and report normal transitions within the application. Examples: &quot;Server is ready&quot;, &quot;Loading dimension...&quot;</td>
</tr>
<tr>
<td>WARN</td>
<td>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: &quot;Attempt to add a new group exceeded the limit on group creation&quot;</td>
</tr>
<tr>
<td>ERROR</td>
<td>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: &quot;Error connecting to remote machine&quot;, &quot;Failed to create a cube&quot;</td>
</tr>
<tr>
<td>FATAL</td>
<td>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.</td>
</tr>
</tbody>
</table>

**TM1 Loggers**

TM1 loggers represent the different areas or subcomponents of TM1 that produce log messages.

For example, dimensions and cubes produce messages when loading and the server reports messages when starting and stopping.

You can control the quantity and focus of the 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: Customer support can provide you with guidance and assistance to enable individual TM1 loggers.
**Logging Properties File**

The logging properties file, tm1s-log.properties, enables you to control which message levels are logged for the different subcomponents of TM1.

The tm1s-log.properties file is a text file that contains the parameters for configuring logging for a specific TM1 server.

Each TM1 server uses its own logging 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 make changes to a properties file in real-time and the TM1 server will dynamically read the changes and adjust the logging.

**Note:** The TM1 local server only checks the logging properties file, tm1s-log.properties, when the local server starts up. For details, see “Enabling and Disabling Transaction Logging” on page 59.

A sample logging properties file is provided in the directory of each sample TM1 database that you install. Sample TM1 databases are installed in the following locations:

- **Windows** system: \install_dir\samples\tm1\sample_dir
- **UNIX** system: \install_dir/samples/tm1/sample_dir

Where `sample_dir` is the directory for a sample TM1 database. For example:

- **Planning Sample Data:** \install_dir\samples\tm1\PlanSamp
- **Sample Data:** \install_dir\samples\tm1\SData

Here is an example of the logging properties file:

```properties
Log4j.rootLogger=INFO, S1
Log4j.logger.TM1=INFO
# 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=5MB
#Specify the max filesize
log4j.appender.S1.MaxFileSize=10MB
#Specify the max backup index
log4j.appender.S1.MaxBackupIndex=3
```

**Configuring and Enabling Server Message Logging**

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

**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 Message Level” on page 65 and “Configuring Logging Output” on page 66.
2. Place the logging properties file, tm1s-log.properties, into the same directory where the configuration file, Tm1s.cfg, is located for the TM1 server you want to monitor. The location of the Tm1s.cfg file is typically the data directory of the TM1 server, 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 logging properties file into this directory.

After the tm1s-log.properties file is placed in the same directory as the Tm1s.cfg file, the server will start logging based on the settings configured in the logging 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 logging properties file, so all settings in a local server's logging 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 Message Level**

Use the following two statements in the logging properties file as the standard configuration to log all INFO level messages 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."

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

```log4j
log4j.rootLogger=DEBUG, S1
log4j.logger.TM1=DEBUG
```

**Configuring Logging Output**

Logging output is configured in the appender section of the logging properties file with the following parameters.

- MemorySize
- MaxFileSize
- MaxBackupIndex
- TimeZone

Each parameter is described in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| MemorySize    | 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. Default value is 5 MB. Use the following format: 

```log4j
log4j.appender.S1.MemorySize=Size Units
```

where:
- Size is the numeric value for the memory size.
- Units can be KB, or MB.

For example, to set the MemorySize to 5 MB, enter the following:

```log4j
log4j.appender.S1.MemorySize=5 MB
```

| MaxFileSize   | Specifies the maximum file size that the log file is allowed to take up on disk. Default size is 100 MB. Uses the following format: 

```log4j
log4j.appender.S1.MaxFileSize=SizeUnits
```

where:
- Size is the numeric value for the file size.
- Units can be KB, MB, or GB.

For example, to limit the log file size to 10 MB, enter the following:

```log4j
log4j.appender.S1.MaxFileSize=10 MB
```

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 the MaxBackupIndex option.
## Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| MaxBackupIndex       | A numeric value that specifies how many backup files are kept before the oldest log file is erased.  
                       | Default value is 20.  
                       | • 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.  
                       | The backup process repeats each time the MaxFileSize is reached, overwriting the oldest backup file so that only the newest backup files remains. |
| TimeZone             | Specifies the timezone for the message timestamps in the log file. Uses the following format:  
                       | log4j.appender.S1.TimeZone=Zone  
                       | where Zone can be set to either GMT or Local.  
                       | • 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. |

## 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.

<table>
<thead>
<tr>
<th>Message Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Thread ID     | ID number of the thread in the TM1 server that generated the logging event.  
<pre><code>                   | Example: 2488                                                               |
</code></pre>
<p>| Message Level | Severity level of the message being reported: DEBUG, INFO, WARN, ERROR, or FATAL. |</p>
<table>
<thead>
<tr>
<th>Message Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date and Time</td>
<td>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 &quot;Configuring Logging Output&quot; on page 66. Reported in the format: yyyy-mm-dd hh:mm:ss,millseconds. Example: 2006-10-02 16:49.01,925</td>
</tr>
<tr>
<td>TM1 Logger Name</td>
<td>Name of the TM1 sub-component that generated the message or activity. Example: Start time: Tue Dec 05 2006 11:02:47</td>
</tr>
<tr>
<td>Message Text</td>
<td>Pre-defined text message that describes the error or activity being reported.</td>
</tr>
</tbody>
</table>

3. 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**
1. Click **Edit, Find** or press CTRL+F.
   The Find dialog box opens.
2. 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**
1. Select the line of text you want to copy.
2. 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.

**Client Logging**
As of IBM Cognos TM1 version 9.5.1, 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:
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's system machine. The TM1p.ini file must be in the same folder.

Be sure that location is also set in the \log4j.append.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.append.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.append.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.append.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.

Using the Audit Log

The TM1 audit log monitors changes to metadata, such as modifications to dimensions, views and subsets.

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?

You can query and view the audit log using the Audit Log window, available in TM1 Server Explorer.

Note: By default, audit logging is disabled and must be manually enabled for each server you want to monitor. For details, see “Configuring Audit Logging” on page 72.

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.

The objects are:
- Applications
- Chores
- Clients
- Cubes
- Dimensions
- Elements
A common set of events, such as object created or deleted, are logged for all these objects. Additionally, events are logged that are specific to each object. For example, the execution of a process is an event that is logged only for process objects.

**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
- 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 would be 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 would display 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 would be reported in the audit log for a manual sort change to the account dimension:

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

The Audit Log window would display 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 the *IBM Cognos TM1 Installation and Configuration Guide*.

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.

<table>
<thead>
<tr>
<th>Message Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Date and time of the event.</td>
</tr>
<tr>
<td>User</td>
<td>TM1 client (user) or scheduled chore that was responsible for causing the event.</td>
</tr>
<tr>
<td>Event Type/Description</td>
<td>Brief description of the event.</td>
</tr>
<tr>
<td>Object Type</td>
<td>Type of TM1 object associated with the event.</td>
</tr>
<tr>
<td>Object Name</td>
<td>Name of the TM1 object associated with the event.</td>
</tr>
<tr>
<td>Details</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

**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.
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.

To export audit log data, complete the following steps.

**Procedure**

1. Click **Export** in the Results toolbar.
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**.
Monitoring Server Performance Using Control Cubes

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.

The following control cubes are populated during performance monitoring. For details, see [Appendix B, “Control Cubes,” on page 145].

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

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 details, see Appendix B, "Control Cubes," on page 145 and Appendix C, "Control Dimensions," on page 159.

   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.

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 standalone 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.

With the exception of 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=
adminsvrsslexportkeyid=
```

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 77.

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdminHost</td>
<td>Machine name or IP address of the computer running the Admin Server. Default value is localhost.</td>
</tr>
<tr>
<td>LogFile</td>
<td>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. Format is: logfile=\path-to-log-file where \path-to-log-file must specify the complete path, filename, and file extension. A default filename and file extension is not provided.</td>
</tr>
<tr>
<td>LogPeriod</td>
<td>Specifies the time interval between updates being written to the log file. This value should be a multiple of the Refresh time parameter. The format for this parameter is: logperiod=nn where nn is the number of elapsed seconds between updates to the log file. Setting this to zero disables logging. 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. The default value is 2 seconds.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LogAppend</td>
<td>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. Valid values are T and F as shown in the following examples: If you set LogAppend=T, log data is appended to the log file specified by the LogFile parameter. If you set LogAppend=F, the existing log file will be over-written. The default value is F, which overwrites the existing log file.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Time interval between refreshing of the TM1 Top display. Format is: refresh=nn where nn is the time interval in seconds. Default value is 2 seconds.</td>
</tr>
<tr>
<td>ServerName</td>
<td>TM1 server name. This is the ServerName from the server configuration file, Tm1s.cfg.</td>
</tr>
</tbody>
</table>

**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.”

**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 77.
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:

tmltop.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 mnn
- -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:

tmltop.exe -servername sdata -refresh 5

Note: Use quotes for parameter values that include spaces, as follows:

tmltop.exe -servername "planning sample"

To show a list of available parameters, use the /? option as follows:

tmltop.exe /?

For more details on TM1 Top parameters, see "Configuring the Tm1top.ini File" on
page 77.

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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
<td>The thread ID of the underlying system thread in the TM1 server.</td>
</tr>
</tbody>
</table>
| **User Name** | The name of the user or process that started the thread. This can be one of three possible thread types.  
  - **User Thread** - Name of an actual user that is logged into TM1. A value of ??? means the user is currently in the login stage.  
  - **Chore Thread** - A chore running on the TM1 server. The name of the chore thread is displayed in the following format: C:chore-name  
  - **System Thread** - A TM1 system process running on the TM1 server. Displayed in the following format: Th:name of system thread  
    - 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** | The current processing state of a specific thread. This value can be one of the following values:  
  - Idle  
  - Run  
  - Commit  
  - Rollback  
  - Wait  
  - Login  
  For more information, see "Understanding TM1 Thread Processing States" on page 83. |
| **Function** | The current API function that the thread is executing.  
  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. |
### Field Name Description

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj Lock Status</td>
<td>Lock status of the last object that was locked. Displayed in the format: R# IX# W#</td>
</tr>
<tr>
<td></td>
<td>Where:</td>
</tr>
<tr>
<td></td>
<td>• R, IX, and W represent the different lock modes on the object. For more information, see “Understanding TM1 Lock Modes” on page 83.</td>
</tr>
<tr>
<td></td>
<td>• # is the number of locks for each specific lock mode.</td>
</tr>
<tr>
<td>Note:</td>
<td>: The name of the object is displayed in the State field.</td>
</tr>
<tr>
<td>User Lock Status</td>
<td>Total lock status and count for all objects being used by a thread. Displayed in the format: R# IX# W#</td>
</tr>
<tr>
<td></td>
<td>Where:</td>
</tr>
<tr>
<td></td>
<td>• R, IX, and W represent the different lock modes on the objects. For more information, see “Understanding TM1 Lock Modes” on page 83.</td>
</tr>
<tr>
<td></td>
<td>• # is the total number of locks, across all objects, for each specific lock mode.</td>
</tr>
<tr>
<td>Time</td>
<td>The total time, in seconds, that the current API function or chore process has been processing.</td>
</tr>
<tr>
<td></td>
<td>The name of the API function or chore process is displayed in the Function field.</td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Sandbox</td>
<td>Name of the currently active sandbox.</td>
</tr>
<tr>
<td>Total Sandbox Memory</td>
<td>The Total Sandbox Memory being consumed for all sandboxes by user.</td>
</tr>
<tr>
<td>Number of Sandboxes</td>
<td>Number of sandboxes for this user.</td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the sandbox in the queue.</td>
</tr>
<tr>
<td>Node Name</td>
<td>Name of the node for the sandbox in the queue.</td>
</tr>
<tr>
<td>Status</td>
<td>Date, time, and state when the status was taken for the sandbox in the queue.</td>
</tr>
<tr>
<td>Time in Queue</td>
<td>Length of time sandbox has been in the queue before it was processed.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Lock Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td><strong>R-lock</strong> (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.</td>
</tr>
<tr>
<td>IX</td>
<td><strong>IX-lock</strong> (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.</td>
</tr>
<tr>
<td>W</td>
<td><strong>W-lock</strong> (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.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>Thread is currently not processing.</td>
</tr>
</tbody>
</table>
## State Description

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
</table>
| Run     | Indicates the thread is actively running. Also displays the lock mode and object being accessed by the thread in the following format:  
Run: Lock Mode-(Object Name)-Object Type  
Where:  
- Lock Mode is one of the possible object lock modes; R, IX, or W. For more information, see “Understanding TM1 Lock Modes” on page 83  
- 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. |
| 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    | Thread is currently waiting for a specific condition to complete. Displayed in the format:  
Wait: Wait Status - (Object Name) - Object Type  
Where:  
- Wait Status can be one of the following conditions:  
  - 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.  
- 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. |
| 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:
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 the following types of threads, determined by the current state of the thread and the function the thread is currently processing:

- **Threads in a Run state** - TM1 can cancel only threads with a state of Run that are running functions involving either TI processing, constructing a view, or calculating a view.
- **Threads in a Wait state** - Any thread that is in a Wait state can be cancelled.

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, as described below.

**Understanding the Cancel Process**

The Cancel process cancels a running or waiting thread.

- If a thread has a Run state, and the process involves TI processing, constructing a view, or calculating a view, then the thread can be cancelled. 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.

### Command Table

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Displays TM1 Top normal (thread status) mode.</td>
</tr>
<tr>
<td>S</td>
<td>Displays TM1 Top sandbox mode.</td>
</tr>
<tr>
<td>Q</td>
<td>Displays TM1 Top queue mode.</td>
</tr>
<tr>
<td>W</td>
<td>Writes the current TM1 Top information to a file.</td>
</tr>
<tr>
<td>X</td>
<td>Exits the TM1 Top utility.</td>
</tr>
<tr>
<td>H</td>
<td>Displays help text.</td>
</tr>
<tr>
<td>V</td>
<td>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.</td>
</tr>
<tr>
<td>C</td>
<td>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.</td>
</tr>
<tr>
<td>K</td>
<td>Cancels a sandbox submission in the job queue for a specific job ID number. Requires administrator login with the V (Verify) command.</td>
</tr>
</tbody>
</table>

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.
• If a thread is in the Wait state, the thread is cancelled in two steps. First, the cancel command releases the thread lock request, and then the cancel command cancels the thread processing.

**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.

---

**Using TM1 Performance Counters**

TM1 performance counters are a collection of values that are continuously updated to provide realtime monitoring of TM1 server performance.

These values are incremented and decremented in realtime 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 Note about Running TM1 Performance Counter Tools
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

Note: To see a complete, updated list of available TM1 performance counters, with descriptions, use the Microsoft Windows Performance Monitor as described in "Adding and Viewing TM1 Performance Counters in the Microsoft Windows Performance Monitor" on page 89.

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 TM1PerfMon
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
```

**Note:** 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
```

**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
```
The title fields in the display include:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter Name</td>
<td>Group name and counter name for each TM1 performance counter.</td>
</tr>
<tr>
<td></td>
<td>Displayed in the format: GroupName:CounterName</td>
</tr>
<tr>
<td></td>
<td>For example: Threads:Threads Created</td>
</tr>
<tr>
<td>Value</td>
<td>Numeric value for the TM1 performance counter at the specific output date and time.</td>
</tr>
</tbody>
</table>

**Exiting TM1 PerfMon**
You can exit the TM1 PerfMon while the utility is still looping.

**Procedure**
Press CTRL+Break at any time.

**Viewing TM1 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**

**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 as shown in the following figure.

Capturing Core Dumps for TM1 Server Crash Analysis

You want to ensure that everything is in place in case you experience a TM1 server crash.

It is imperative that the appropriate debugging tools are configured correctly and the correct files are collected and sent to IBM Cognos support and/or engineering for analysis.
Debugging Tools for Windows

You can use either ADPlus or Dr. Watson as a debugging tool to collect server crash information on a Windows system.

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.mspx.

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 cd to the Debugging Tools for Windows directory. The default directory on a 32-bit system is C:\Program Files\Debugging Tools for Windows (x86).
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.
Monitoring the TM1 Server:

If you are monitoring the server, just continue your daily activity.

When the server crashes, three dump files (.dmp) are created in the Crash_Mode_DateStamp_TimeStamp subdirectory. The only one IBM needs to debug the server crash is the ...2nd chance_AccessViolation... file.

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
dump 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 (C:\Program Files\Debugging Tools for Windows (x86) if you
   accepted the default installation directory on a 32-bit system.) 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/40b9eb1-3578-48f9-93e1-7ca6760c1c151033.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. (C:\Program Files\Debugging Tools for Windows (x86) if you accepted the default installation directory on a 32-bit system.)

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
   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. Again, this is C:\Program Files\Debugging Tools for Windows (x86) if you accepted the default installation directory on a 32-bit system.

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 the following line at the command prompt on the client computer:

```
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 drwtsn32, 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 drwtsn32 set as your Debugger.

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.

- If you are running TM1 Top, collect the TM1 Top log file. The location and name of the log file are determined by the logfile parameter to the Tm1top.ini configuration file.
  
  The Tm1top.ini configuration file is located in the same directory as the TM1 Top executable file (Tm1top.exe), which is usually install dir\bin. If you accepted the default TM1 installation directory, the full path is C:\Program Files\IBM\cognos\tm1\bin.

- DO NOT collect the TM1 transaction log (Tm1s.log) unless specifically requested by IBM support.

Transmitting Files to Cognos

Once all required files are collected, you must deliver the files to Cognos support for analysis.

There are several options for delivering the files.

The quickest way to deliver files to IBM is to upload them to the IBM FTP server. You can access the IBM FTP server from a command prompt or from a Web browser.

**Uploading from a Command Prompt**

You can upload files to the IBM FTP server from a command prompt.

**Procedure**

1. cd to the directory where the file you want to upload resides.
2. At the command prompt, enter ftp testcase.boulder.ibm.com.
3. When prompted for a username, enter anonymous.
4. When prompted for a password, enter your email address.
   
   The Virtual user anonymous logged in message confirms that you are connected to the FTP server.
5. cd to the /toibm/im directory.
6. At the command prompt, enter bin to set the upload mode to binary.
7. At the command prompt, enter put followed by the name of the file you want to upload. For example, to upload a file named MyDumpFiles.zip, enter put MyDumpFiles.zip.

**Uploading from a Web Browser**

You can upload files to the IBM FTP server from a Web browser.

**Procedure**

2. Log in with user name anonymous. No password is required for an anonymous login.
3. Click the toibm folder.
4. Click the im folder.
5. Click Browse and navigate to the file you want to upload.
6. Click Upload file (binary).

**Uploading to Your FTP Server**

Alternatively, you can upload the files to your own FTP server and contact IBM support with instructions for retrieving the files.

You should use your own FTP server only if you cannot establish a connection to the IBM FTP server.
Chapter 8. Cognos TM1 Tools and Utilities

Use the following tools and utilities with IBM Cognos TM1.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM1Top</td>
<td>Command line tool for monitoring TM1 server activity. See &quot;Using the TM1 Top Utility&quot; on page 77.</td>
</tr>
<tr>
<td>Cognos TM1 Operations Console</td>
<td>Web-based, enhanced version of TM1Top with added features and a graphical user interface. See the IBM Cognos TM1 Operations Console Guide.</td>
</tr>
<tr>
<td>TM1RunTI</td>
<td>Command line interface tool that initiates a Cognos TM1 TurboIntegrator (TI) process from within any application capable of issuing operating system commands. See &quot;Using TM1RunTI&quot; in the TM1 TurboIntegrator Guide.</td>
</tr>
<tr>
<td>Synchronized()</td>
<td>Cognos TM1 TurboIntegrator (TI) function used in a TurboIntegrator script to force serial execution of a designated set of TurboIntegrator processes. See &quot;Serializing TurboIntegrator processes using synchronized() in the IBM Cognos TM1 TurboIntegrator Guide.</td>
</tr>
<tr>
<td>tm1xfer</td>
<td>Command line utility used to compress Cognos 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 &quot;tm1xfer.&quot;</td>
</tr>
<tr>
<td>ODBC_test</td>
<td>The odbc_test tool is used to diagnose and test the Cognos TM1 ODBC connection on UNIX. See odbc_test tool” on page 106.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;directory&gt;</td>
<td>Specifies the path and name of the TM1 database directory that contains the files to be modified and compressed for transfer.</td>
</tr>
</tbody>
</table>
Table 3. tm1xfer parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;file&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>-v</td>
<td>Displays the version number of the tm1xfer tool.</td>
</tr>
<tr>
<td>-h</td>
<td>Displays comprehensive help documentation for the application.</td>
</tr>
<tr>
<td>-?</td>
<td>Displays a short help dialog, which includes usage syntax and version number.</td>
</tr>
<tr>
<td>no parameters</td>
<td>Displays the short help dialog (same as -?).</td>
</tr>
</tbody>
</table>

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.
To move the database files to an 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 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 warming message and prompt 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+\text{xxxx}, where \text{xxxx} is the UTF-16 code point for that character.

tm1xfer only processes files with version number 9.1 or higher. Earlier file versions are ignored. To process earlier version files, first convert them to version 9.1 or later.

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 a 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 vice versa) 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.

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
<th>Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>This locale is not supported by your Java™ installation</td>
<td>The filesystem encoding is not supported by this program.</td>
<td>Update your Java installation so it supports the filesystem encoding.</td>
</tr>
<tr>
<td>&lt;arg&gt; does not exist</td>
<td>The file or directory name provided in the command line argument is a file or directory that does not exist.</td>
<td>Ensure that the file or directory name on the command line is correct.</td>
</tr>
<tr>
<td>Cannot resolve &lt;file&gt;: &lt;message&gt;</td>
<td>There was a problem resolving the full path to the file.</td>
<td>Use the specific information in the message to determine the source of the problem.</td>
</tr>
<tr>
<td>Cannot resolve &lt;base_dir&gt;: &lt;message&gt;</td>
<td>There was a problem resolving the full path to the base directory.</td>
<td>Use the specific information in the message to determine the source of the problem.</td>
</tr>
<tr>
<td>Cannot serialize out list of files: &lt;message&gt;</td>
<td>There was an error writing this Java object to disk.</td>
<td>Use the specific information in the message to determine the source of the problem.</td>
</tr>
<tr>
<td>Cannot open &lt;filename&gt; for reading: &lt;message&gt;</td>
<td>tm1xfer does not have permission to read this file in the database directory.</td>
<td>Check your filesystem permissions. Use the specific information in the message to determine the source of the problem.</td>
</tr>
<tr>
<td>Error writing to zip file: &lt;message&gt;</td>
<td>There was a problem writing to the zip file.</td>
<td>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.</td>
</tr>
<tr>
<td>Error closing zip file: &lt;message&gt;</td>
<td>There was a problem closing the zip file. Usually this problem resolves itself when tm1xfer exits.</td>
<td>Use the specific information in the message to determine the source of the problem.</td>
</tr>
<tr>
<td>Error opening zip file: &lt;message&gt;</td>
<td>tm1xfer checks for a missing file; however, Java requires a FileNotFoundException log message.</td>
<td>Use the specific information in the message to determine the source of the problem.</td>
</tr>
<tr>
<td>&lt;filename&gt;: Not a valid archive</td>
<td>The passed argument exists but it is not a valid archive that was created by tm1xfer</td>
<td>Ensure the file name on the command line is correct.</td>
</tr>
<tr>
<td>Error reading zip file: &lt;message&gt;</td>
<td>There was an error reading the zip file. The file may have been corrupted or there may be a permission problem.</td>
<td>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.</td>
</tr>
<tr>
<td>Error reading &lt;filename&gt;</td>
<td>There was an unspecified error while reading the current file</td>
<td>Use the specific information in the message to determine the source of the problem.</td>
</tr>
</tbody>
</table>
Table 4. Error Messages for tm1xfer (continued)

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
<th>Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error interpreting path: &lt;message&gt;</td>
<td>There was a problem resolving this path</td>
<td>Use the specific information in the message to determine the source of the problem.</td>
</tr>
<tr>
<td>File read error: &lt;message&gt;</td>
<td>There was a problem reading this file.</td>
<td>Ensure that tm1xfer has read permission for the file.</td>
</tr>
<tr>
<td>Error reading Application Entries dimension</td>
<td>There was an error reading</td>
<td>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.</td>
</tr>
</tbody>
</table>

**tm1xfer usage notes**

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 IBM Cognos TM1 install does not add the Cognos TM1 server bin directory to the users PATH variable. Since you normally will run this tool in the directory containing the Cognos 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 Cognos 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 Cognos TM1 database (and application data) to a different server and (b) deploying a Cognos 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 Cognos TM1 server should be shut down before running tm1xfer to pack up the files. tm1xfer sometimes detects that the Cognos TM1 server was not properly shut down, and will prompt the user to determine whether they should continue or not.

**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**

```bash
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:

```ini
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 9. Administering IBM Cognos TM1 Web

You can configure IBM Cognos TM1 to work over the Web.

Changing Your Password in Cognos TM1 Web

Users can change their own IBM Cognos TM1 Web passwords on the login screen.

Procedure

1. On the Cognos TM1 Web login screen, enter your user name and existing password.
2. Click the Change Password check box.
3. Click Login.
   - The Change User Password page opens.
4. Enter your new password in the New Password box.
5. Enter your new password a second time in the Confirm New Password box.
6. Click OK to save your new password and continue with the login procedure.

Modifying Cognos TM1 Web Configuration Parameters

The tm1web_config.xml file is an XML file that contains configuration parameters for IBM Cognos TM1 Web.

As of Cognos TM1 Web version 10.2, the new tm1web_config.xml file replaces the web.config file from previous Cognos TM1 Web versions.

The parameters in this file control the following IBM Cognos TM1 Web features.
- View node
- Cube Viewer page size
- Number of sheets to export from a Cube Viewer
- IBM Cognos TM1 Web startup and appearance settings

Cognos TM1 Web Configuration Parameters

The configuration parameters for IBM Cognos TM1 Web are stored in the tm1web_config.xml file.

The tm1web_config.xml file is located in the following location:

<TM1 install location>/webapps/tm1web/WEB-INF/configuration/

The following parameters are available.

**CubeViewerPageSize**

Specifies the number of rows to fetch in a page of cubeviewer.

See "Changing the Cube Viewer Page Size" on page 120.

**CubeViewerColumnPageSize**

Specifies the number of columns to fetch in a page of cubeviewer.

See "Changing the Cube Viewer Page Size" on page 120.
GzipCompressionEnabled
Determined if the web server responses will be compressed. Valid values are true/false.

IntegratedSecurityModuleName
Specifies the name of the login module in the file pointed to by the java.security login configuration file.

NavTreeDisplayServerView
Specifies whether to display the Server View node in the navigation tree. Valid values are Y and N.
See “Displaying or Hiding the Views Node in the Navigation Pane” on page 120.

NavTreeHidden
Determines if the navigation panel will be displayed after a user logs in.
See “NavTreeHidden Parameter” on page 118.

NavTreeCollapsedOnStart
Determines if the navigation panel will be collapsed or expanded after a user logs in.
See “NavTreeCollapsedOnStart Parameter” on page 118.

HideTabBar
If set to true, multiple tabs will not be displayed.
See “HideTabBar Parameter” on page 119.

HideWebsheetToolBar
If set to true, all websheet toolbars will not be displayed.
See “HideWebsheetToolBar Parameter” on page 119.

HideCubeviewerToolBar
If set to true, all Cubeviewer toolbars will not be displayed.
See “HideCubeviewerToolBar Parameter” on page 119.

HomePageObject
If set, the object of type of Websheet, Cubeviewer or URL will be displayed after a user logs in.
See “Configuring a Global Homepage for All Users” on page 114.

MaximumSheetsForExport
Maximum number of sheets allowed to Export.
See “Setting the Maximum Number of Sheets to Export from a Cube Viewer” on page 120.

AdminHostName
If set, users will not be asked to enter a value for Admin Host during login.
See “Configuring the Cognos TM1 Web Login Page using AdminHostName and TM1ServerName parameters” on page 112.

AdminHostPort
If set, the client will try to use this port instead of the default Admin Host port.
AdminHostSSLPort
If set, the client will try to use this port instead of the default Admin SSL Host port.

TM1ServerName
If set, users will not be asked to select a TM1 Server to connect to during login.
See "Configuring the Cognos TM1 Web Login Page using AdminHostName and TM1ServerName parameters" on page 112.

CubeviewerStringWrap
Settings for string cell wrapping in the cubeviewer.
See "Wrapping string values in cube views" on page 121.

RecalcOnDataValidationChange
Specifies whether the default recalculation behavior will be overridden when changing the value of a data validation list.
If set to true, a recalculation will be triggered when a value in a data validation list is changed.
If set to false, a recalculation will not be triggered when a value in a data validation list is changed.

RecalcOnPicklistChange
Specifies whether the default recalculation behavior will be overridden when changing the value of a picklist.
If set to true, a recalculation will be triggered when a value in a picklist is changed.
If set to false, a recalculation will not be triggered when a value in a picklist is changed.

Editing the Cognos TM1 Web configuration file
You can edit the IBM Cognos TM1 Web configuration file to configure different parameters.

The Cognos TM1 Web configuration file is an xml file and should be opened only with an XML-type editor. Opening it using a regular text editor such as Microsoft Wordpad can result in incorrect characters being added that may corrupt the file.

As of Cognos TM1 Web version 10.2, the new tm1web_config.xml file replaces the web.config file from previous Cognos TM1 Web versions.

Procedure
1. Locate and open the tm1web_config.xml file in the following location:
   <TM1 install location>\webapps\tm1web\WEB-INF\configuration\</TM1 install location>

   Note: The tm1web_config.xml file is an xml file and should be opened only with an XML-type editor. Opening it using a regular text editor such as Microsoft Word Pad can result in incorrect characters being added that may corrupt the file.
2. Edit the parameters and save your changes.
3. Log in to IBM Cognos TM1 Web to see the result of your edits.
Configuring the Cognos TM1 Web Login Page using AdminHostName and TM1ServerName parameters

The AdminHostName and TM1ServerName parameters control whether or not the IBM Cognos TM1 Web login page prompts the user to enter values for the TM1 Admin Host and TM1 server.

If you set a value for either of these parameters in the tm1web_config.xml file, then the login process uses the specified value and does not prompt the user for this information.

AdminHostName Parameter

This parameter specifies the name of the Admin Host on which a TM1 Admin Server is running. Edit the AdminHostName parameter in the tm1web_config.xml file using the following format:

```xml
<add key="AdminHostName" value="HostName"/>
```

where HostName can be one of the following values:

- If HostName is blank (default value), then the login page displays the Admin Host prompt.
- If HostName is set to the name of a valid TM1 Admin Host, then IBM Cognos TM1 Web uses that Admin Host for the login process and does not prompt the user.

TM1ServerName Parameter

This parameter sets the name of the TM1 server. Edit the TM1ServerName parameter in the tm1web_config.xml file using the following format:

```xml
<add key="TM1ServerName" value="ServerName"/>
```

where ServerName can be one of the following values:

- If ServerName is blank (default value), then the TM1 server prompt is displayed on the IBM Cognos TM1 Web login page.
- If ServerName is set to a valid TM1 server name, then the login page does not display a prompt for either the Admin Host or the TM1 server.

After the user enters a valid User Name and Password, IBM Cognos TM1 Web will login to the TM1 server specified by the TM1ServerName parameter in the tm1web_config.xml file.

For example, the TM1ServerName parameter could be set to planning sample, as shown in the following code.

```xml
<add key="TM1ServerName" value="planning sample"/>
```

Configuring a Custom Homepage for IBM Cognos TM1 Web

You can configure a custom homepage for IBM Cognos TM1 Web to display a Websheet, cube view, or a URL after users have successfully logged into IBM Cognos TM1 Web. This homepage can provide users with a starting point for accessing and working with TM1 data.

A homepage can be configured globally for all IBM Cognos TM1 Web users or assigned individually for different users or sets of users. For example, if you...
configure the homepage option to display an HTML file or other type of web page, then you can provide users with instructions, tasks, links, or any other content that can be displayed in a web page.

If a homepage is configured, it displays on the first tab in IBM Cognos TM1 Web and cannot be closed by users. When configured, a Home link is displayed in the header area of IBM Cognos TM1 Web that allows users to easily return to the homepage.

An IBM Cognos TM1 Web homepage can be configured in one of the following two ways:

- **Different homepage for different IBM Cognos TM1 Web users** - Use the Client Settings dialog in TM1 Architect and Server Explorer to configure a startup homepage for different clients (users) of IBM Cognos TM1 Web.

- **Global homepage for all IBM Cognos TM1 Web users** - Use the HomePageObject parameter in the tm1web_config.xml file to configure a homepage that applies globally to all IBM Cognos TM1 Web users.

**Note:** Any homepage assignment you make with the Client Settings dialog can override the global setting in the tm1web_config.xml file if you set AllowOverwrite=true in the HomePageObject parameter of the tm1web_config.xml file.

**Configuring Different Homepages for Individual Users**

The Client Settings dialog, in Architect and Server Explorer, configures a startup homepage for different IBM Cognos TM1 Web clients (users).

For example, you can assign one homepage for IBM Cognos TM1 Web users in the Sales department and another homepage for users in the Finance department.

**Note:** You can use the Client Settings dialog to assign homepages for specific users, over-riding the global homepage setting for the HomePageObject parameter in the tm1web_config.xml file.

**Procedure**

1. In Architect or Server Explorer, right click on the server and select **Security, Clients/Groups**. The Clients/Groups dialog opens.
2. Click **Settings**. The Client Settings dialog opens.
3. Select the client from the **Current Client** list for which the homepage setting will apply.
4. Enter a Websheet, cube view, or URL for the homepage as follows:
   - To display a URL, type the URL address, including the http:// protocol, into the Homepage box. You can enter a URL for either a website or an individual file.
   - To select a Websheet or cube view as the homepage, click **Browse**. The Select an IBM Cognos TM1 Web Homepage dialog opens where you can select a reference to a Websheet or cube view from the Application tree.
     After selecting a Websheet or cube view reference, click **OK** to return to the Client Settings dialog.
5. Select the settings that control the appearance of the Navigation pane.
Note: The Navigation pane settings you set here will only apply if the corresponding parameter in the tm1web_config.xml file is set to AllowOverwrite=true. For details, see “Configuring IBM Cognos TM1 Web Startup and Appearance Settings” on page 117.

The available settings for controlling the appearance of the Navigation pane include:

- **Include the Navigation Pane** - Determines if the Navigation pane is displayed or not displayed when the selected client logs in to IBM Cognos TM1 Web.
- **Open pane on Login** - Sets the Navigation pane to display in the expanded mode when the selected client logs in to IBM Cognos TM1 Web.
- **Close pane on Login** - Sets the Navigation pane to display in its minimized mode when the selected client logs in to IBM Cognos TM1 Web.
- **Save Client's Navigation Pane Settings** - Determines if the personal settings for the Navigation pane are saved when the client logs out of IBM Cognos TM1 Web.

6. Select one of the options from the Apply To list to configure which client or clients will be able to view the homepage.

   - **Current Client** - Applies the homepage setting for only the client selected in the current Client list.
   - **Selected Clients** - Enables the Select button so you can open the Subset Editor to select a collection of clients that will use the same homepage setting.
   - **All Clients** - Applies the same homepage setting to all TM1 clients.

If you choose **Selected Clients**, and then click **Select**, the Subset Editor opens so you can select a subset of TM1 clients that can use the homepage.

Use the Subset Editor to select a subset of clients and then click **OK** to return to the Client Settings dialog. The number of clients selected in the Subset Editor is summarized in the Client Settings dialog.

7. Click **Apply Settings** to configure the homepage for the client or clients that you selected in the Apply To list.

8. Repeat steps 4, 5, 6, and 7 to configure a homepage for a different set of TM1 clients.

9. Click **OK** to close the Client Settings dialog.

You have now configured a homepage for IBM Cognos TM1 Web. The selected IBM Cognos TM1 Web clients will see the assigned homepage the next time they successfully log in to IBM Cognos TM1 Web.

### Configuring a Global Homepage for All Users

The HomePageObject parameter, in the tm1web_config.xml file, enables a global homepage that displays for all IBM Cognos TM1 Web users.

**Note:** You can override the global HomePageObject parameter by using the Client Settings dialog to assign different homepage's for individual Cognos TM1 users. For details, see “Configuring Different Homepages for Individual Users” on page 113.

The HomePageObject parameter works for three types of objects:

- Cubeviewer
- Websheet
- URL
Using the HomePageObject Parameter:

How to use the HomePageObject parameter.

The HomePageObject parameter uses the following format:

```xml
<add key="HomePageObject" value="ObjectPath;Type=ObjectType;Description=ObjectTitle;AllowOverwrite=true"/>
```

where:
- `ObjectPath` is the path to the Websheet, cube view, or URL object that you want to open. The exact format of the path depends on the type of object.
- `ObjectType` is the keyword for the object you want to open; websheet, cubeviewer, or URL.
- `ObjectTitle` is a brief title you assign to the object that displays in the title bar of the web browser and on the homepage tab in IBM Cognos TM1 Web.
- `AllowOverwrite` can be set to a value of true or false as follows:
  - If you set `AllowOverwrite=true` then the HomePageObject parameter can be overridden by setting a different homepage for individual clients using the Client Settings dialog in Architect and Server Explorer.
  - If you set `AllowOverwrite=false` then the HomePageObject parameter applies globally to all TM1 users and can not be individually configured with the Client Settings dialog in Architect and Server Explorer.

The following sections describe using the HomePageObject parameter for Websheets, cube views, and URLs.

Setting a Global IBM Cognos TM1 Web Homepage to a Cube View:

Use the following format to set a cube view as the homepage for IBM Cognos TM1 Web.

```xml
value=CubeName$ViewName$Status
```

where the following arguments are separated by $ characters:
- `CubeName` is the name of cube to which the view belongs.
- `ViewName` is the name of the cube view to display.
- `Status` is the public or private status of the cube view.

**Note:** You must include a value of either PUBLIC or PRIVATE to correctly identify the specific cube view that you want to open.

For example, to open a public view named Price from the SalesCube:

```xml
&lt;add key="HomePageObject" value="SalesCube$$Price$$Public;Type=cubeviewer;Description=MyStartCube;AllowOverwrite=true"/>
```
Setting a Global IBM Cognos TM1 Web Homepage to a Websheet:

You can assign a Websheet as the IBM Cognos TM1 Web homepage, depending on how the Excel file was added to TM1.

Opening a Websheet that references an Excel file outside of TM1:

You can open a Websheet that references an Excel file.

Procedure

Use the format:

value="WebsheetPath"

where WebsheetPath is the location and name of the Excel file. This can be either a path for a local file, or a UNC path for a file located on a network.

For example, to set a UNC network path for Websheet:

value=//MySystem/Samples/classic_slice.xls

Results

The complete HomePageObject parameter looks like this:

<add key="HomePageObject" value="/MySystem/Samples/classic_slice.xls;Type=websheet;Description=MyWebsheet;AllowOverwrite=true"/>

Opening a Websheet object that was uploaded to the TM1 server:

You can open a Websheet object that was uploaded.

Procedure

1. In Server Explorer, use the Properties pane to find the TM1 assigned name for the uploaded Excel file.

2. Set the value parameter using the following format:

value="TM1://ServerName/blob/PUBLIC/\}Exterals\ TM1 _Filename"

where:

- ServerName is the name of the TM1 server where the Excel file is located.
- TM1 _Filename is the name that TM1 assigned to the uploaded Excel file.
For example:

```
value="TM1://sdata/blob/PUBLIC/\Externals\Report_2006.xls_20070123212746.xls"
```

The complete HomePageObject parameter line looks like this:

```
<add key="HomePageObject" value="TM1://sdata/blob/PUBLIC/\Externals\Report_2006.xls_20070123212746.xls;Type=websheet;Description=My Uploaded Websheet;AllowOverwrite=true" />
```

Setting a Global IBM Cognos TM1 Web Homepage to a URL:

You can set the HomePageObject parameter to a URL.

Use this format:

```
value="URL_Path"
```

Where `URL_Path` can point to a web site or an individual web page file.

For example:

- To set the homepage to a URL that points to a file:

  ```
  <add key="HomePageObject" value="homepage.html;Type=URL;Description=MyStart Page;AllowOverwrite=true" />
  ```

- To set the homepage to a URL that points to a web site:

  ```
  <add key="HomePageObject" value="http://www.ibm.com;Type=URL;Description=IBM;AllowOverwrite=true" />
  ```

Displaying reminders to save data

Use the SuppressPleaseSaveDialog parameter to suppress or display the reminders to save data prior to performing an action such as data spreading or changes in pick-lists that trigger a recalculation.

When the parameter is set to 1 in the web.config file, the dialog boxes reminding you to save your data do not display and any changed data is automatically submitted prior to performing an action such as data spreading or changes in pick-lists that trigger a recalculation. If the SuppressPleaseSaveDialog parameter is not present in web.config the reminder dialog boxes do not display.

To restore the display of those dialog boxes when data is changed but not yet saved, you can manually set the SuppressPleaseSaveDialog parameter to 0.

- 0 - Indicates that the dialog boxes will not be suppressed
- 1 - Indicates that the dialog boxes will be suppressed

**Note:** The web.config file that ships with Cognos TM1 9.5.2 FP2 has the SuppressPleaseSaveDialog parameter set to 0 to maintain behavior consistent with previous 9.5.2 releases.

Configuring IBM Cognos TM1 Web Startup and Appearance Settings

You can control the appearance of the Navigation pane, tab bar, and Websheet and Cubeviewer toolbars when users log in to IBM Cognos TM1 Web.

These parameters are located in the tm1web_config.xml file and apply globally to all users of IBM Cognos TM1 Web.
Note: For details on using the HomePageObject parameter to set a custom homepage, see "Configuring a Custom Homepage for IBM Cognos TM1 Web" on page 112.

**NavTreeHidden Parameter**

The NavTreeHidden parameter determines if the Navigation pane displays when users log in to IBM Cognos TM1 Web.

This can be helpful if you are displaying a custom homepage for users and you want to completely hide the Navigation pane.

The NavTreeHidden parameter uses the following format in the tm1web_config.xml file:

```xml
<add key="NavTreeHidden" value="false;AllowOverwrite=true"/>
```

where:

value can be either true or false

- If set to false, the Navigation pane will be displayed when user's log in to IBM Cognos TM1 Web.
- If set to true, the Navigation pane will not be displayed when user's log in to IBM Cognos TM1 Web.

AllowOverwrite can be set to true or false as follows:

- If you set AllowOverwrite=true, the NavTreeHidden parameter is assigned globally to all users, but can be overridden for individual clients using the Client Settings dialog in Architect and Server Explorer.
- If you set AllowOverwrite=false, the NavTreeHidden parameter applies globally to all TM1 users and can not be overridden for individual clients using the Client Settings dialog in Architect and Server Explorer.

**NavTreeCollapsedOnStart Parameter**

The NavTreeCollapsedOnStart parameter determines if the Navigation pane will be minimized or expanded when users log in. If collapsed, a small vertical bar displays to provide the user with a way to restore the pane.

The NavTreeCollapsedOnStart parameter uses the following format in the tm1web_config.xml file:

```xml
<add key="NavTreeCollapsedOnStart" value="false;AllowOverwrite=true"/>
```

where:

value can be either true or false.

- If value is set to false, the Navigation pane will be expanded and display in its default mode when user's log in to IBM Cognos TM1 Web.
- If value is set to true, the Navigation pane will be collapsed when user's log in to IBM Cognos TM1 Web.

AllowOverwrite can be set to true or false as follows:

- If you set AllowOverwrite=true, the NavTreeCollapsedOnStart parameter is assigned globally to all users, but can be overridden for individual clients using the Client Settings dialog in TM1 Architect and Server Explorer.
If you set AllowOverwrite=false, the NavTreeCollapsedOnStart parameter applies globally to all TM1 users and cannot be overridden for individual clients using the Client Settings dialog in TM1 Architect and Server Explorer.

**HideTabBar Parameter**

The HideTabBar parameter determines if IBM Cognos TM1 Web can display multiple tabs when a user opens multiple IBM Cognos TM1 Web objects, or if only one view is displayed.

This can be useful if you want to limit users to one view at a time.

The HideTabBar parameter uses the following format in the tm1web_config.xml file:

```xml
<add key="HideTabBar" value="false;AllowOverwrite=true"/>
```

where value can be either true or false.

- If value is set to false, multiple tabs can be displayed. This is the default behavior of IBM Cognos TM1 Web.
- If value is set to true, multiple tabs are not displayed and only one object can be opened at a time.

The AllowOverwrite option is not currently used for this parameter.

**HideWebsheetToolBar Parameter**

The HideWebsheetToolBar parameter determines if the Websheet toolbar is displayed when users open a Websheet.

The HideWebsheetToolBar parameter uses the following format in the tm1web_config.xml file:

```xml
<add key="HideWebsheetToolBar" value="false;AllowOverwrite=true"/>
```

where value can be either true or false.

- If value is set to false, the Websheet toolbar will display in IBM Cognos TM1 Web.
- If value is set to true, the Websheet toolbar will not display in IBM Cognos TM1 Web.

The AllowOverwrite option is not currently used for this parameter.

**HideCubeviewerToolBar Parameter**

The HideCubeviewerToolBar parameter determines if the Cubeviewer toolbar is displayed when users open a cube view.

The HideCubeviewerToolBar parameter uses the following format in the tm1web_config.xml file:
<add key="HideCubeviewerToolBar" value="false;AllowOverwrite=true" />

where value can be either true or false.

- If value is set to false, the Websheet toolbar will display in IBM Cognos TM1 Web.
- If value is set to true, the Websheet toolbar will not display in IBM Cognos TM1 Web.

The AllowOverwrite option is not currently used for this parameter.

**Displaying or Hiding the Views Node in the Navigation Pane**

You can display or hide the Views node in the Navigation pane.

**Procedure**

1. Edit \tm1web_config.xml\ in the IBM Cognos TM1 Web virtual directory.
2. Locate the NavTreeDisplayServerView, which controls the display of the Server View node. The default value, Y, displays the Views node in the Navigation pane.
   
   <!--NavTreeDisplayServerView: Y/N - Wether to display "Server View" node in navigation tree -->
   
   <add key="NavTreeDisplayServerView" value="Y" />
3. To hide the Views node, change the NavTreeDisplayServerView value to N.
4. Save tm1web_config.xml.
5. Log in to IBM Cognos TM1 Web.

Now the Navigation pane displays without the View node.

**Changing the Cube Viewer Page Size**

You can change the number of rows and columns displayed in the Cube Viewer of IBM Cognos TM1.

By default, Web Cube Viewer displays pages of TM1 data with 20 columns and 100 rows, and includes the dimensions list in the row count.

**Procedure**

1. Edit \tm1web_config.xml\.
2. Locate the following code:

   CubeViewerRowPageSize
   CubeViewerColumnPageSize
3. Change the value for the row and/or column page size.
4. Save tm1web_config.xml.
5. Log in to IBM Cognos TM1 Web.

For example, if you set the row page size to 10, the Cube Viewer displays nine rows of data, plus the row of dimensions.

**Setting the Maximum Number of Sheets to Export from a Cube Viewer**

By default, the maximum number of sheets you can export from a Cube Viewer to a printer is 100. You can configure IBM Cognos TM1 Web to export more sheets.
Procedure
1. Edit tm1web_config.xml.
2. Locate the following code:
   ```
   MaximumSheetsForExport
   ```
3. Change the value for the maximum number of sheets to export.
4. Save tm1web_config.xml.
5. Log in to IBM Cognos TM1 Web.

Wrapping string values in cube views
Use CubeviewerStringWrap to set the parameters used when viewing string element cells in a Web Cube View.

To control the way a view is displayed and wrapped, set the values using the CubeviewerStringWrap parameter and save the web configuration file. Cells that are not displayed are still editable in a scrollable area by clicking in the wrapped region.

**Enabled**
- Turn wrapping of string cells in this view on or off. When set to "False" the column width is as wide as the longest string for any row in the current view. Set to "True" by default to turn on wrapping using these default parameters.

**MinCharactersToWrap**
- Set the minimum number of characters needed before wrapping. For instance, string values with less than 50 characters will not wrap within a cell. Set to 50 by default.

**MaxDisplayCharacters**
- Set the maximum number of characters to display within the string cell. The cell may contain more than this number of characters, but they will only be displayed when double-clicking on the cell. If the MinCharactersToWrap is 50 and the MaxDisplayCharacters is 200, string cells containing 200 or more characters will consume approximately 4 lines. Set to 200 by default.

**WidthOfWrapCell**
- Set the number of characters used in the wrapped portion of the display. Set to 240 by default.

Use the following format in the tm1web_config.xml file (the following listing has a return in it for clarity but you should not enter a return).

```xml
<add key="CubeviewerStringWrap" value="Enabled=true;MinCharactersToWrap=50;MaxDisplayCharacters=200;WidthOfWrapCell=240" />
```

**Remember:** CubeviewerStringWrap does not apply to Websheets.

Using IBM Cognos TM1 Web Logging
IBM Cognos TM1 Web administrators can use the tm1web.log file for status and troubleshooting of Cognos TM1 Web. The severity levels in the log file help organize messages.
IBM Cognos TM1 Web log file

The logging process for IBM Cognos TM1 Web records activity and error messages for the program into the `tm1web.log` file.

Administrators can use this log file for status and troubleshooting of IBM Cognos TM1 Web. The severity levels in the log files help organize messages.

The `tm1web.log` file is an ASCII text file that you can open in any text editor, such as Microsoft Windows Notepad.

Log file name and location

Log files are stored in the following location:

`<TM1 installation location>\webapps\tm1web\WEB-INF\logs`

The current or most recent file is named `tm1web.log`.

Older files are saved and time-stamped with the following name and date format:

`tm1web.log.yyyy-mm-dd`

For example:

`tm1web.log.2013-03-21`.

Message Severity Levels for IBM Cognos TM1 Web Logging

The logging process for IBM Cognos TM1 Web categorizes log messages into three severity levels.

These levels are also used in the logging properties file to configure logging to a specific level.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUG</td>
<td>Detailed, technical messages that are useful when TM1 customer support or development engineering need to debug the application. When logging is configured to this level, DEBUG, INFO, and ERROR messages are logged.</td>
</tr>
<tr>
<td>INFO</td>
<td>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.</td>
</tr>
<tr>
<td>ERROR</td>
<td>An error condition of which you should be aware. Action should be taken to fix or report the issue to TM1 customer support. When logging is configured to this level, only ERROR messages are logged.</td>
</tr>
</tbody>
</table>
Configuring and enabling IBM Cognos TM1 Web logging

You can change the logging message level for IBM Cognos TM1 Web logging.

Logging properties are stored in the log4j.properties file in the following location:

<TM1 install location>/webapps/tm1web/WEB-INF/configuration

Logging for Cognos TM1 Web is configured and enabled by default when the program is installed.

Attention: The default web logging configuration is intended for everyday use and does not typically require adjustment. For assistance if you need to configure the logging properties for troubleshooting purposes, contact IBM Cognos Customer Support.

The following is a sample of the logging properties file.

# System logging settings
log4j.rootLogger=ERROR, TextFile
log4j.logger.com.ibm.cognos=ERROR
log4j.logger.com.ibm.cognos.org=ERROR
log4j.logger.com.ibm.cognos.perf=ERROR
log4j.logger.com.ibm.cognos.tm1=ERROR

log4j.appender.Console=org.apache.log4j.ConsoleAppender
log4j.appender.Console.layout.ConversionPattern=%d [%t] %-5p (%x) %c - %m%n

log4j.appender.TextFile=org.apache.log4j.DailyRollingFileAppender
log4j.appender.TextFile.File=logs/tm1web.log
log4j.appender.TextFile.DatePattern=.yyyy-MM-dd
log4j.appender.TextFile.layout=org.apache.log4j.PatternLayout
log4j.appender.TextFile.layout.ConversionPattern=%d [%t] %-5p (%x) %c - %m%n

log4j.appender.XMLFile=org.apache.log4j.DailyRollingFileAppender
log4j.appender.XMLFile.File=logs/tm1web_log.xml
log4j.appender.XMLFile.DatePattern=.yyyy-MM-dd
log4j.appender.XMLFile.layout=org.apache.log4j.xml.XMLLayout

You can adjust various logging level and output options in this file.

The message level is indicated by:

log4j.logger.logger_name=message_level

The log file name is indicated by:

log4j.appender.appender_name.File=location

Attention: By default, the log file is created beneath the root of your web server. As such, it could be accessible by unauthorized individuals. Consider setting the File parameter to write the log file to a secure location. The parameter can accept a relative or literal path.

Procedure

1. Open the log4j.properties file in a text editor, such as Microsoft Windows Notepad.
2. Locate and edit the line you want to adjust.
   For example, change the message level to one of the valid values; DEBUG, INFO, or ERROR.
3. Save and close the file.

**Viewing the IBM Cognos TM1 Web Log File**

The IBM Cognos TM1 Web installation configures IBM Cognos TM1 Web logging to write messages to the `tm1web.log` file in the `<TM1 Web_install>\WEB-INF\logs\` directory. You can open and view the file with a standard text editor.

**About this task**

If you installed IBM Cognos TM1 Web to the default installation location, then the `tm1web.log` file is located in the following directory:

C:\Program Files\IBM\cognos\tm1_64\webapps\tm1web\WEB-INF\logs

For backup purposes, a copy of the `tm1web.log` file is renamed and saved on a daily basis using the following naming convention:

`tm1web.log.<year>-<mm>-<dd>`

For example, `tm1web.log.2013-10-17`.

**Procedure**

1. Locate the `tm1web.log` file in the `<TM1 Web_install>\WEB-INF\logs\` directory.
2. Open and view the file with a text editor, such as Microsoft Windows Notepad.

**Results**

Error messages are arranged in the following format:

```
Date Time Error_level Logger_name Error_message
```

Where:

- **Date Time** - Date and time in format `yyyy-mm-dd hh:mm:ss`.
  For example `2013-05-02 16:48:57,439`
- **Error_level** - message level (DEBUG, INFO, ERROR)
- **Logger_name** - the sub component name. Example: `Cognos.TM1.Web.PageTM1WebpageUtils`
- **Error_message** - the message text.

**Microsoft Excel .xls worksheets**

IBM Cognos TM1 Web version 10.2 uses the Open XML file formats for Microsoft Excel worksheets created using Excel 2007 or later.

If you are using existing Microsoft Excel files in the older .xls format, use the Cognos TM1 conversion tool to convert the files. If your original file contained macros, the Cognos TM1 conversion tool converts the original file into a macro-enabled .xslm file, otherwise it is converted into a standard .xlsx file.
The Convert Excel files to OpenXML Excel format option in Cognos TM1 Architect Server Explorer converts a single .xls worksheet or all worksheets in a folder. Only administrative users have this option available. The conversion renames the files to preserve as many links as possible after the conversion. Some links and action buttons need to be updated depending on permissions that may have changed as a result of the move to cell-based security that occurred in version 10.2.0.

In some cases, the Named Ranges from the original file could be renamed in the converted file during the conversion process.

By default a backup of the pre-converted worksheets is saved. By default a log file is also generated.

Converting a .xls worksheet to .xlsx
The one-time conversion of .xls worksheets results in an Open XML format Excel file that can be used in TM1 Web.

Procedure
1. In IBM Cognos TM1 Architect Server Explorer, right-click the worksheet or folder you want to convert. Only Microsoft Excel .xls files will be converted regardless of other files that may be in the folder.
2. Select Convert Excel file to OpenXML format.
3. By default a backup of the pre-converted .xls file and a log is created in the directory locations displayed. You can browse to identify new locations for these files, if you prefer.
4. When the conversion is completed, the window lists the number of files found and completed and the location of the log text file that was generated.
5. You may need to re-establish links to some files or action buttons. The change to cell-based security means some files may not have the correct permissions to work without some manual adjustments.
Chapter 10. Understanding Cube Dependency

IBM Cognos 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:

![Diagram showing Base Cube and Dependent Cube]

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

TM1 9.4.1 FP3 introduced a change to the dependency mapping to make TM1 more stable and to perform better.

There is an artifact of this change that users upgrading to TM1 9.4.1 FP3 and above should understand. In previous releases, this dependency map between cubes was established when rules were executed that referenced other cubes. However, these dependencies were cleared when there was any invalidation between the cubes and these dependencies would again be re-established once the rule was run to establish the dependency.

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 TM1 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. In the 9.5 series, other enhancements are available to better establish dependencies without any manual intervention.

### Clearing Cube Dependencies

The following table describes the TM1 events or actions that destroy a cube's dependency.

<table>
<thead>
<tr>
<th>Event</th>
<th>Dependency Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Shutdown</td>
<td>Cube dependency lists are destroyed.</td>
</tr>
<tr>
<td>Rule Compile (Note: any dimension update causes a rule compile.)</td>
<td>Eliminates only those cubes where rules have been deleted or modified.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Event</th>
<th>Dependency Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Startup (not available for TM1 9.4.1 FP3 HS7)</td>
<td>Calculated Feeders will create dependencies.</td>
</tr>
<tr>
<td>Server Startup + Persistent Feeders (not available for TM1 9.4.1 FP3 HS7)</td>
<td>Persistent Feeders also store dependency maps for each cube.</td>
</tr>
<tr>
<td>Rule Compile</td>
<td>Calculated Feeders for the particular cube rule will create dependencies.</td>
</tr>
<tr>
<td>Rule Calculation</td>
<td>Create dependencies based which rule is executed by the query.</td>
</tr>
<tr>
<td>TI Function</td>
<td>TI Function can create dependencies between cubes.</td>
</tr>
</tbody>
</table>

### Using AddCubeDependency to Establish Dependencies

The AddCubeDependency function is used to create a manual dependency between two cubes.

AddCubeDependency is a TurboIntegrator function, valid only in TurboIntegrator processes.

This function creates a manually-created dependency between Cube A and Cube B.

**Syntax:** `AddCubeDependency('CubeA','CubeB');`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CubeA</td>
<td>The name of the base cube.</td>
</tr>
<tr>
<td>CubeB</td>
<td>The name of the dependent cube.</td>
</tr>
</tbody>
</table>

Cube B relies on a rule that is dependent on Cube A.
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:

```
   Adding cube dependency: cube
   
   "}ElementSecurity_}Cubes" depends on cube "}CubeSecurity"
```

The TurboIntegrator command for this would be:

```
AddCubeDependency('}CubeSecurity','}ElementSecurity_}Cubes');
```
Chapter 11. Replicating Cubes

This section describes how you can replicate cubes from one IBM Cognos TM1 server to another, and synchronize the updates across the copied cubes.

Replication Overview

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 Cognos TM1 should only be performed by members of the ADMIN group. Members of the DataAdmin and SecurityAdmin groups 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 bi-directional 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 details, see “Configuring the Tm1s.cfg File to Support Replication” on page 136.

- **Length of directory path and cube name** - The total length of the pathname for the target TM1 server’s data directory and the name of the cube you are replicating can not exceed the Windows pathname limit of approximately 256 characters. If this limit is exceeded, due to a long pathname 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 bi-directional synchronization, transaction logging must be enabled for all the related cubes on both the source and target servers.

  TM1 uses these log files to keep track of the changes made to the source and mirror cubes. To verify that transaction logging is enabled, see “Enabling and Disabling Transaction Logging” on page 59 in the IBM Cognos TM1 Operation Guide.

- **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 wanted the Measures dimension for a replicated cube to be set on the target server, you would have to 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 DataAdmin and SecurityAdmin groups 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 "Synchronization Process" on page 141.

Admin Server Considerations

When you establish a replication connection, both the source server and the target server must be registered on the same Admin Server. If the two servers do not share the same Admin Server, the replication and synchronization results are unpredictable.

For details on verifying that the source and target servers register with the same Admin Server, see "Configuring the Tm1s.cfg File to Support Replication" on page 136.

For details on the Admin Server, see "TM1 Admin Server" on page 19.
Setting 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 "Replication Process" on page 138.
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 "Synchronization Process" on page 141.

Configuring the Tm1s.cfg File to Support Replication
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.

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.

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
```

For more details about configuration parameters, see the IBM Cognos TM1 Installation and Configuration Guide.

Maintaining Replication Connections
In order to replicate cubes, you must first create a replication connection between two servers.

Creating 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 “Synchronization Process” on page 141.

Modifying 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**.

Deleting 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 “Synchronization Process” on page 141.

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.

Replicating a Cube

When replicating a cube, the total length of the pathname for the target TM1 server's data directory and cube name can not exceed the Microsoft Windows pathname limit of approximately 256 characters. If this limit is exceeded, due to a long pathname 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 139.
5. Click OK.
   TM1 replicates the cube, as described in “What Happens when TM1 Replicates a Cube” on page 140.

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.

Specifying 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.

Specifying 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.

Specifying 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.

Specifying 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.

Specifying 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.

### 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 59 in the IBM Cognos TM1 Operation Guide.

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).

### Synchronizing 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**.

### Scheduling Synchronization

You can create a chore to automatically synchronize data at a regular interval using TurboIntegrator.

For details on TurboIntegrator, see the IBM Cognos TM1 TurboIntegrator Guide.
**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 \( \leftrightarrow \) 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.

**Synchronizing 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 the *IBM Cognos TM1 Installation and Configuration Guide*. 
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.

Table 5. Keyboard shortcuts

<table>
<thead>
<tr>
<th>Action</th>
<th>Shortcut key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform the command for an active command button</td>
<td>Enter</td>
</tr>
<tr>
<td>Close an object or pane that is closable</td>
<td>Ctrl+W</td>
</tr>
<tr>
<td>Go to the first item or object; Go Home</td>
<td>Home</td>
</tr>
<tr>
<td>Go to the last item or object; Go to the End</td>
<td>End</td>
</tr>
<tr>
<td>Move forward through the panes of the application</td>
<td>F8</td>
</tr>
<tr>
<td>Move backward through the panes of the application</td>
<td>Shift+F8</td>
</tr>
<tr>
<td>Move the focus to the Application Bar (blue dot)</td>
<td>Alt+F10</td>
</tr>
<tr>
<td>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</td>
<td>Tab</td>
</tr>
<tr>
<td>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</td>
<td>Shift+Tab</td>
</tr>
<tr>
<td>Toggle on or off</td>
<td>Space bar</td>
</tr>
<tr>
<td>Move to the next option button and select it</td>
<td>Right arrow, Down arrow</td>
</tr>
<tr>
<td>Move to the previous option button and select it</td>
<td>Up arrow, Left arrow</td>
</tr>
<tr>
<td>Open and display the contents of a drop-down list</td>
<td>Down arrow</td>
</tr>
<tr>
<td>Close an open drop-down list</td>
<td>Esc</td>
</tr>
<tr>
<td>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</td>
<td>Down arrow</td>
</tr>
<tr>
<td>Move to the previous selectable node</td>
<td>Up arrow</td>
</tr>
</tbody>
</table>
### Table 5. Keyboard shortcuts (continued)

<table>
<thead>
<tr>
<th>Action</th>
<th>Shortcut key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expand the current selection if it is not expanded. If the node is</td>
<td>Right arrow and plus sign</td>
</tr>
<tr>
<td>expanded, go to the first child node</td>
<td></td>
</tr>
<tr>
<td>Collapse the current selection if it is expanded. If the node is</td>
<td>Left arrow and minus sign</td>
</tr>
<tr>
<td>collapsed, go to the parent node before the current selection</td>
<td></td>
</tr>
<tr>
<td>Expand the children menu items</td>
<td>Right arrow</td>
</tr>
<tr>
<td>Collapse the children menu items</td>
<td>Left arrow</td>
</tr>
<tr>
<td>Open a context menu</td>
<td>Right-click key (Mozilla Firefox); Shift+F10 (Microsoft Internet</td>
</tr>
<tr>
<td>Close an open context menu</td>
<td>Esc</td>
</tr>
<tr>
<td>Scroll down</td>
<td>Down arrow or Page Down</td>
</tr>
<tr>
<td>Scroll up</td>
<td>Up arrow or Page Up</td>
</tr>
<tr>
<td>Move to the next widget in the tab index order at the same level in</td>
<td>Tab</td>
</tr>
<tr>
<td>the canvas</td>
<td></td>
</tr>
<tr>
<td>Move to the previous widget in the tab index order at the same level</td>
<td>Shift+Tab</td>
</tr>
<tr>
<td>in the canvas</td>
<td></td>
</tr>
<tr>
<td>Add a database instance</td>
<td>Alt+N</td>
</tr>
<tr>
<td>Upload a configuration file</td>
<td>Alt+U</td>
</tr>
<tr>
<td>Download a configuration file</td>
<td>Alt+I</td>
</tr>
<tr>
<td>Tile vertically</td>
<td>Alt+Q</td>
</tr>
<tr>
<td>Tile horizontally</td>
<td>Alt+W</td>
</tr>
<tr>
<td>Box tile</td>
<td>Alt+P</td>
</tr>
<tr>
<td>Refresh the tree</td>
<td>Alt+R</td>
</tr>
</tbody>
</table>

### IBM and accessibility

See the IBM Accessibility Center for more information about the commitment that IBM has to accessibility.

[IBM Accessibility Center](www.ibm.com/able)
Appendix B. Control Cubes

IBM Cognos TM1 uses control cubes.

The IBM Cognos 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 159.

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 the IBM Cognos TM1 Developer Guide.

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 the IBM Cognos TM1 Developer Guide.

**Dimensions**

The ChoreSecurity control cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chores</td>
<td>Each chore currently defined on the TM1 server.</td>
</tr>
<tr>
<td>Groups</td>
<td>Each user group defined for the TM1 server.</td>
</tr>
</tbody>
</table>

**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 Developer Guide.

Dimensions

The CubeSecurity control cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubes</td>
<td>Cubes available on the TM1 server.</td>
</tr>
<tr>
<td>Groups</td>
<td>Groups defined for the TM1 server.</td>
</tr>
</tbody>
</table>

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 the IBM Cognos TM1 Developer Guide.

**Dimensions**

The |DimensionSecurity control cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dimensions</td>
</tr>
<tr>
<td></td>
<td>Groups</td>
</tr>
</tbody>
</table>

**|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 the IBM Cognos TM1 Developer Guide.

Dimensions

An |ElementSecurity_DimensionName cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DimensionName</td>
<td>All elements of the dimension are present.</td>
</tr>
<tr>
<td></td>
<td>Groups</td>
</tr>
</tbody>
</table>

|ProcessSecurity

The |ProcessSecurity control cube stores security definitions for all TurboIntegrator processes on the TM1 server.

Dimensions

The |ProcessSecurity cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processes</td>
</tr>
<tr>
<td></td>
<td>Groups</td>
</tr>
</tbody>
</table>

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 the IBM Cognos TM1 Developer Guide.

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

**Dimensions**

The }ClientsGroups control cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>}Clients</td>
<td>Clients currently defined for the TM1 server.</td>
</tr>
<tr>
<td>}Groups</td>
<td>Groups defined for the TM1 server.</td>
</tr>
</tbody>
</table>

**}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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>}Clients</td>
<td>Clients currently defined for the TM1 server.</td>
</tr>
<tr>
<td>}ClientProperties</td>
<td>Properties that can be set for TM1 clients, including Password, PasswordExpirationDays, and MaximumPorts.</td>
</tr>
</tbody>
</table>

For complete details on this dimension, see [Appendix C, “Control Dimensions,” on page 159](#).
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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>}Connections</td>
<td>Replication connections defined for the TM1 server.</td>
</tr>
<tr>
<td>}ConnectionProperties</td>
<td>Properties that can be set for replication connections, including User, Password, Syncstar and Syncplanet.</td>
</tr>
</tbody>
</table>

For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 159.

}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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cubes</td>
</tr>
<tr>
<td></td>
<td>CubeProperties</td>
</tr>
<tr>
<td></td>
<td>For complete details on this dimension, see Appendix C, “Control</td>
</tr>
<tr>
<td></td>
<td>Dimensions,” on page 159</td>
</tr>
</tbody>
</table>

|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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dimensions</td>
</tr>
<tr>
<td></td>
<td>DimensionProperties</td>
</tr>
<tr>
<td></td>
<td>For complete details on this dimension, see Appendix C, “Control</td>
</tr>
<tr>
<td></td>
<td>Dimensions,” on page 159</td>
</tr>
</tbody>
</table>

|DimensionAttributes

The |DimensionAttributes control cube stores attribute 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 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.
Dimensions

The |DimensionAttributes cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dimensions</td>
</tr>
<tr>
<td></td>
<td>DimensionAttributes</td>
</tr>
</tbody>
</table>

}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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DimensionName</td>
<td>All elements of the dimension are present.</td>
</tr>
<tr>
<td></td>
<td>ElementAttributes_DimensionName</td>
</tr>
<tr>
<td></td>
<td>For details on defining element attributes, see the IBM Cognos TM1 Developer Guide.</td>
</tr>
</tbody>
</table>

}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 the IBM Cognos TM1 Developer Guide.
**Dimensions**

The ]HierarchyProperties cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>]Dimensions</td>
<td>Contains a list of all the dimension names in the TM1 server.</td>
</tr>
<tr>
<td>]Hierarchies</td>
<td>Includes only one element for assigning named levels to the selected dimension.</td>
</tr>
</tbody>
</table>
| ]HierarchyProperties | Includes the following two types of elements  
  * 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 165.

**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 on a minute-by-minute basis. You can then browse these cubes to analyze server performance.

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 details on performance monitoring, see Chapter 4, “Remote Cognos TM1 Server Operations,” on page 25.

**]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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>}StatsByClient</td>
<td>This dimension contains the following measures:</td>
</tr>
<tr>
<td></td>
<td>Message Count</td>
</tr>
<tr>
<td></td>
<td>Message Bytes</td>
</tr>
<tr>
<td></td>
<td>Request Count</td>
</tr>
<tr>
<td></td>
<td>Elapse Time</td>
</tr>
<tr>
<td></td>
<td>Bytes/Message</td>
</tr>
<tr>
<td></td>
<td>Measures are described in <a href="#">Appendix C, “Control Dimensions,” on page 159</a></td>
</tr>
<tr>
<td>}PerfClients</td>
<td>Each client on the server, plus the consolidated element Clients</td>
</tr>
<tr>
<td></td>
<td>Total, which is a consolidation of all clients.</td>
</tr>
<tr>
<td>}TimeIntervals</td>
<td>See a description of the }TimeIntervals dimension in</td>
</tr>
<tr>
<td></td>
<td><a href="#">Appendix C, “Control Dimensions,” on page 159</a></td>
</tr>
</tbody>
</table>

**}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, and stored views.
Dimensions

The |StatsByCube cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StatsByCube</td>
</tr>
<tr>
<td></td>
<td>PerfCubes</td>
</tr>
</tbody>
</table>

| |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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PerfClients</td>
</tr>
</tbody>
</table>
### Dimension Elements

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>PerfCubes</td>
<td>Each cube on the server, plus the consolidated element Cubes Total, which is a consolidation of all cubes.</td>
</tr>
<tr>
<td>CubeFunctions</td>
<td>This dimension contains the following numeric elements:</td>
</tr>
<tr>
<td></td>
<td>Cell Update</td>
</tr>
<tr>
<td></td>
<td>Cell Retrieval</td>
</tr>
<tr>
<td></td>
<td>View Calculation</td>
</tr>
<tr>
<td></td>
<td>View Retrieval</td>
</tr>
<tr>
<td></td>
<td>For complete details on this dimension, see Appendix C, &quot;Control Dimensions,&quot; on page 159.</td>
</tr>
<tr>
<td>StatsByCubeByClient</td>
<td>This dimension contains the following measures:</td>
</tr>
<tr>
<td></td>
<td>Count</td>
</tr>
<tr>
<td></td>
<td>Elapse Time (milliseconds)</td>
</tr>
<tr>
<td></td>
<td>For complete details on this dimension, see Appendix C, &quot;Control Dimensions,&quot; on page 159.</td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>StatsForServer</td>
<td>This dimension contains the following measures:</td>
</tr>
<tr>
<td></td>
<td>Number of Connected Clients</td>
</tr>
<tr>
<td></td>
<td>Number of Active Threads</td>
</tr>
<tr>
<td></td>
<td>Memory Used (bytes)</td>
</tr>
<tr>
<td></td>
<td>Memory in Garbage (bytes)</td>
</tr>
<tr>
<td></td>
<td>For complete details on this dimension, see Appendix C, &quot;Control Dimensions,&quot; on page 159.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold</td>
<td>This dimension contains the following elements:</td>
</tr>
<tr>
<td></td>
<td>OriginalVal</td>
</tr>
<tr>
<td></td>
<td>Hold Status</td>
</tr>
<tr>
<td></td>
<td>For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 159.</td>
</tr>
</tbody>
</table>

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 Cognos TM1 uses control dimensions to track performance statistics, administer security, manage clients and groups, and store object attributes and properties.

The following information is provided for each dimension:

- Description of dimension elements
- Control cubes that use the dimension

### 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 Cognos TM1 server.

### ClientProperties

This ClientProperties dimension is used only in the ClientProperties control cube.

The dimension contains the following string elements that correspond to the following properties that can be defined for clients on the IBM Cognos TM1 server.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaximumPorts</td>
<td>Indicates the maximum number of ports (connections) that an individual client can open on the TM1 server.</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Stores and displays the client password in an encrypted form.</td>
</tr>
<tr>
<td>PasswordExpirationDays</td>
<td>Indicates the number of days for which a given client's password remains valid.</td>
</tr>
<tr>
<td>PasswordLastTimeUpdated</td>
<td>Indicates the last time (GMT) the password for a given client was updated. The time format is yyyyymmddhhss.</td>
</tr>
<tr>
<td>STATUS</td>
<td>When a client is logged on to the TM1 server, the STATUS value is ACTIVE. Otherwise the STATUS value is empty.</td>
</tr>
</tbody>
</table>

### Clients

The Clients dimension contains string elements that correspond to the clients defined on the IBM Cognos 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 Cognos TM1 server:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGRATEDSECURITY</td>
<td>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.</td>
</tr>
<tr>
<td>LASTSYNC</td>
<td>Time (GMT) at which the data was last synchronized on the planet server, from which the replication connection was established. The time format is yyyymddhhss.</td>
</tr>
<tr>
<td>LASTSYNCSTAR</td>
<td>Time (GMT) at which the data was last synchronized on the star server for the replication connection. The time format is yyyymddhhss.</td>
</tr>
</tbody>
</table>
| LASTSYNCSTARRECORD    | Time (GMT) at which the most recent synchronization resulted in data moving from the star server to the planet. The time format is yyyymddhhss.  
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, not 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 Cognos TM1 server. This dimension is used only in the ConnectionsProperties 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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell Retrieval</td>
<td>Cells retrieved by a user in a non-stored view. This metric includes calculated cells, such as rules, consolidations, and user-defined calculations.</td>
</tr>
<tr>
<td>Cell Update</td>
<td>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.</td>
</tr>
<tr>
<td>View Calculation</td>
<td>Number of views requested, which can be user requests for Cube Viewer or slices.</td>
</tr>
<tr>
<td>View Retrieval</td>
<td>Number of views requested and displayed.</td>
</tr>
</tbody>
</table>

**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 Cognos TM1 server:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DemandLoad</td>
<td>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.</td>
</tr>
<tr>
<td>Lock</td>
<td>When a cube is locked, this property stores the name of the TM1 client that has locked the cube.</td>
</tr>
<tr>
<td>Logging</td>
<td>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.</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Measures_Dimension</td>
<td>If a measures dimension has been defined for a given cube, this property stores the name of the dimension.</td>
</tr>
<tr>
<td>Replication</td>
<td>For replicated cubes, this property stores the name of the replication connection with which the replicated cube is associated.</td>
</tr>
<tr>
<td>RepSrcName</td>
<td>For replicated cubes, this property stores the name of the source cube.</td>
</tr>
<tr>
<td>RepStatus</td>
<td>If a cube is replicated on the server, the value of RepStatus is Copied, otherwise this property value is empty.</td>
</tr>
<tr>
<td>SyncRule</td>
<td>For replicated cubes, this property indicates whether rules are synchronized when the associated cube is synchronized.</td>
</tr>
<tr>
<td>SyncViews</td>
<td>For replicated cubes, this property indicates whether views are synchronized when the associated cube is synchronized.</td>
</tr>
<tr>
<td>Time_Dimension</td>
<td>If a time dimension has been defined for a given cube, this property stores the name of the dimension.</td>
</tr>
<tr>
<td>VMM</td>
<td>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. If no VMM value is specified the default value is 65,536 bytes. The valid range is 16,384 - 42,934,943,296 bytes.</td>
</tr>
<tr>
<td>VMT</td>
<td>For each cube, this property defines the time threshold, in seconds, beyond which the algorithm that stores TM1 stargate views is triggered. 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. If no VMM value is specified the default value is five seconds. The valid range is 1 - 259,200 seconds.</td>
</tr>
</tbody>
</table>

For details on stargate views, see "About Stargate Views" on page 39.

---

**}Cubes**

The }Cubes dimension contains string elements corresponding to the cubes, including control cubes, on the IBM Cognos TM1 server.

The }Cubes dimension is used only in the }DimensionAttributes control cube.
The `DimensionAttributes` dimension contains numeric elements that correspond to the attributes defined for dimensions on the IBM Cognos TM1 server.

The `DimensionAttributes` dimension is used in the `CubeSecurity` and `CubeProperties` control cube.

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 Cognos TM1 server.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default_Hierarchy</td>
<td>Not currently used.</td>
</tr>
<tr>
<td>DemandLoad</td>
<td>Not currently used.</td>
</tr>
<tr>
<td>Last_Time_Updated</td>
<td>Time (GMT) at which a dimension was last updated. The time format is yyyyymmddhhss.</td>
</tr>
<tr>
<td>Lock</td>
<td>When a dimension is locked, this property stores the name of the TM1 client that has locked the dimension.</td>
</tr>
<tr>
<td>Replication</td>
<td>For replicated cubes, this property stores the name of the replication connection with which the replicated cube is associated.</td>
</tr>
<tr>
<td>RepSrcName</td>
<td>For replicated dimensions, this property stores the name of the source dimension.</td>
</tr>
<tr>
<td>RepStatus</td>
<td>If a dimension is replicated on the server, the value of RepStatus is Copied. Otherwise the RepStatus value is empty.</td>
</tr>
<tr>
<td>SortComponentsSense</td>
<td>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.</td>
</tr>
<tr>
<td>SortComponentsType</td>
<td>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.</td>
</tr>
<tr>
<td>SortElementsSense</td>
<td>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.</td>
</tr>
</tbody>
</table>
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.

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 Cognos 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 Cognos 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.
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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>defaultMember</td>
<td>A string element that stores the name of the default member of the dimension.</td>
</tr>
<tr>
<td></td>
<td>The element name you enter here may filter the dimension when TM1 data is retrieved from an external application like IBM Cognos Report Studio.</td>
</tr>
<tr>
<td></td>
<td>Use the name of the top element in the dimension hierarchy so all the dimension elements are retrieved by default.</td>
</tr>
<tr>
<td>level000 - level020</td>
<td>A set of string elements that store custom names for the hierarchy levels of a dimension.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OriginalVal</td>
<td>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.</td>
</tr>
<tr>
<td>HoldStatus</td>
<td>This string element identifies cells held by UserName in the CubeName cube. 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.</td>
</tr>
</tbody>
</table>

The PerfClients dimension contains numeric elements that correspond to the clients defined on the IBM Cognos TM1 server as well as a consolidation of all clients named Clients Total.

The PerfClients dimension is used in the following performance monitoring cubes:
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 Cognos 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 Cognos TM1 server.

The |Processes dimension is used only in the |ProcessSecurity control cube.

**|StatsStatsByClient**

The |StatsStatsByClient dimension is used in the |StatsByClient control cube.

This dimension contains the following measures:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes/Message</td>
<td>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</td>
</tr>
<tr>
<td>Elapse Time (ms)</td>
<td>Time (in milliseconds) required to process requests to the TM1 server.</td>
</tr>
<tr>
<td>Message Bytes</td>
<td>Number of bytes that have been sent across in the requests.</td>
</tr>
<tr>
<td>Message Count</td>
<td>A message is a request to the TM1 server for a list of objects, such as a list of dimensions.</td>
</tr>
<tr>
<td>Request Count</td>
<td>Measures the number of requests for information from the TM1 server.</td>
</tr>
</tbody>
</table>
The JStatsStatsByCube dimension is used only in the JStatsByCube control cube.

This dimension tracks performance statistics when performance monitoring is enabled, and contains the following elements:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Used for Calculations</td>
<td>This metric measures the memory used in all TM1 calculations, including consolidations, rules, and user-defined calculations.</td>
</tr>
<tr>
<td>Memory Used for Feeders</td>
<td>This metric measures the amount of memory used to feed cells through TM1 rules.</td>
</tr>
<tr>
<td>Memory Used for Input Data</td>
<td>This includes memory allocated to data loaded through TurbolIntegrator or process worksheets. This measures the memory used for all data input other than slice or Cube Viewer input.</td>
</tr>
<tr>
<td>Memory Used for Views</td>
<td>This metric measured the amount of memory, in bytes, used to store views for a given cube on the TM1 server.</td>
</tr>
<tr>
<td>Number of Fed Cells</td>
<td>This is the number of cells in the cube targeted by feeders.</td>
</tr>
<tr>
<td>Number of Populated Numeric Cells</td>
<td>This metric counts all the populated string cells in a given cube. String cells are cells identified by at least one string element.</td>
</tr>
<tr>
<td>Number of Populated String Cells</td>
<td>This metric counts all the populated string cells in a given cube. String cells are cells identified by at least one string element.</td>
</tr>
<tr>
<td>Number of Stored Calculated Cells</td>
<td>This metric counts all stored calculated cells including rules, dimensional consolidations, and user-defined consolidations.</td>
</tr>
<tr>
<td>Number of Stored Views</td>
<td>This metric measures the number of named views of a given cube.</td>
</tr>
<tr>
<td>Total Memory Used</td>
<td>The total amount of memory used by a cube, measured in bytes.</td>
</tr>
</tbody>
</table>
**}StatsStatsByCubeByClient

The }StatsStatsByCubeByClient dimension is used only in the }StatsByCubeByClient control cube.

The dimension contains the following elements:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>For a given cube function, this metric measures the number of functions executed.</td>
</tr>
<tr>
<td>Elapsed Time (ms)</td>
<td>For a given type of cube function, this metric measures the elapsed time required to execute all functions.</td>
</tr>
</tbody>
</table>

**}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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory In Garbage</td>
<td>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.</td>
</tr>
<tr>
<td>Memory Used</td>
<td>The total amount of memory used by the server, in bytes.</td>
</tr>
<tr>
<td>Number of Active Threads</td>
<td>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.</td>
</tr>
<tr>
<td>Number of Connected Clients</td>
<td>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.</td>
</tr>
</tbody>
</table>

**}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 TimeInterval 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.
Some Microsoft Excel events are handled by the IBM Cognos TM1 add-in for Excel.

### List of Events

<table>
<thead>
<tr>
<th>Excel Event Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewWorkbook</td>
<td>This event is handled to track the number of workbooks opened.</td>
</tr>
<tr>
<td>WorkbookOpen</td>
<td>This event is handled to track the number of workbooks opened and also to set the state of TM1 toolbars and charts.</td>
</tr>
<tr>
<td>WorkbookBeforeSave</td>
<td>This event is handled to detect modified status of dependent TM1 Objects and to prompt save on such objects.</td>
</tr>
<tr>
<td>WorkbookBeforeClose</td>
<td>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.</td>
</tr>
<tr>
<td>WorkbookActivate</td>
<td>This event is handled to correctly set the status of TM1 toolbars and menu bars.</td>
</tr>
<tr>
<td>WorkbookAddinUninstall</td>
<td>This event is handled to remove TM1 toolbars and menu bars from a workbook.</td>
</tr>
<tr>
<td>SheetActivate</td>
<td>This event is handled to update the TM1 toolbar and menu bar states, as well as to update sheet-specific variables.</td>
</tr>
<tr>
<td>SheetDeactivate</td>
<td>This event is handled to detect sheet deletion.</td>
</tr>
<tr>
<td>SheetSelectionChange</td>
<td>This event is handled to update TM1 menus and toolbar states.</td>
</tr>
<tr>
<td>SheetChange</td>
<td>This event is handled to update the TM1 database with the changes made to a particular cell.</td>
</tr>
<tr>
<td>SheetBeforeDoubleClick</td>
<td>This event is handled to post TM1 -specific dialog boxes on particular TM1 cells.</td>
</tr>
<tr>
<td>SheetBeforeRightClick</td>
<td>This event is handled to add and modify Excel context menu.</td>
</tr>
</tbody>
</table>
Notices

This information was developed for products and services offered worldwide.

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 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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.
Trademarks

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 at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

- 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.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- 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.

Microsoft product screen shot(s) used with permission from Microsoft.
Index

Special characters
- h 21
  .cfg file 19, 22
  .ini file 30
  .rej file 25, 34, 59
  .xlsx worksheet 124
  CellSecurity_CubeName control cube 145
  Chores control dimension 159
  ChoreSecurity control cube 146
  ClientProperties control cube 150
  ClientProperties control dimension 159
  Clients control dimension 159
  ClientSecurity control cube 146
  ClientsGroups control cube 149
  ConnectionProperties control cube 151
  ConnectionProperties control dimension 160
  Connections control dimension 161
  CubeFunctions control dimension 161
  CubeProperties control cube 151
  CubeProperties control dimension 161
  Cubes control dimension 162
  CubeSecurity control cube 147
  DimensionAttributes control cube 152
  DimensionAttributes control dimension 163
  DimensionProperties control cube 152
  DimensionProperties control dimension 163
  Dimensions control dimension 164
  DimensionSecurity control cube 148
  ElementAttributes_DimensionName control cube 153
  ElementAttributes_DimensionName control dimension 164
  ElementSecurity_DimensionName control cube 148
  Groups control dimension 164
  Hierarchies control dimension 164
  HierarchyProperties control cube 153
  HierarchyProperties control dimension 165
  Hold control dimension 165
  Hold_UserName_CubeName control cube 158
  PerfClients control dimension 165
  PerfCubes control dimension 166
  Processes control dimension 166
  ProcessSecurity control cube 149
  StatsByClient 76
  StatsByClient control cube 155
  StatsByCube 76, 156
  StatsByCube control cube 156
  StatsByCubeByClient 76
  StatsByCubeByClient control cube 156
  StatsForServer 76
  StatsForServer control cube 157
  StatsStatsByClient control dimension 166
  StatsStatsByCube control dimension 167
  StatsStatsByCubeByClient control dimension 168
  StatsStatsForServer control dimension 168
  TimeIntervals control dimension 168

access (continued)
  privileges, groups 45
  privileges, Read 135, 141
  privileges, synchronization 141
  privileges, Write 141
  accessibility 143, 144
  keyboard shortcuts 143
  Add Counters 89
  ADMIN group 47
    assigning passwords 51
  Admin Host 28, 30
    location 22
    resetting 30
    specifying multiple 23
  Admin server 19, 25, 28
    diagram 19
    IP address 19
    name 19
    port number 19
    protocol 19
    status 21
    UNIX 22
    Windows 21
    Windows Startup directory 21
    Windows status 22
  AdminHost parameter 78
    specify location 22
  administration
    tm1web_config.xml 109
  administrative groups 46
  assigning users to groups 50
  Audit log
    configuring 72
    copying data 75
    details 74
    exporting results 75
    find feature 75
    object events 70
    queries 73
    results 74
    system-wide events 70
    understanding events 70
    updating with latest records 72
    using 70
    viewing 72
  Audit log, overview 70
  authentication 43
  integrated login 43
  authentication, LDAP 43

B
  background mode 29
  Block Access to Server Capability Assignment 5
  browser view 39
  bulk load mode 40
cancel TM1 Top 85
Capability assignments 5
cell-level security 45
chain configuration 132
server 132
change password 52, 109
chore
synchronization 142
clearing passwords 52
client
connections 37
disconnecting 38
messages 37
query 60
client logging 68
command prompt
TM1 Top 84
commit TM1 Top 83
configuration parameters
TM1 Web 109
configure
TM1 Top utility 78
configure login page 112
control cube 37, 76, 145

|CellSecurity_CubeName 145 |
|ChoreSecurity 146 |
|ClientProperties 150 |
|ClientSecurity 146 |
|ClientsGroups 149 |
|ConnectionProperties 151 |
|CubeProperties 151 |
|CubeSecurity 147 |
|DimensionAttributes 152 |
|DimensionProperties 152 |
|DimensionSecurity 148 |
|ElementAttributes_DimensionName 153 |
|ElementSecurity_DimensionName 148 |
|HierarchyProperties 153 |
|Hold_UserName_CubeName 158 |
|ProcessSecurity 149 |
|StatsByClient 155 |
|StatsByCube 156 |
|StatsByCubeByClient 156 |
|StatsForServer 157 |
cube and group administration 149
object attribute and property 151
performance monitoring 154
security 145
control dimension 159
|Chores 159 |
|ClientProperties 159 |
|Clients 159 |
|ConnectionProperties 160 |
|Connections 161 |
|CubeFunctions 161 |
|CubeProperties 161 |
|Cubes 162 |
|DimensionAttributes 163 |
|DimensionProperties 163 |
|Dimensions 164 |
|ElementAttributes_DimensionName 164 |
|Groups 164 |
|Hierarchies 164 |
|HierarchyProperties 165 |
|Hold 165 |
|PerfClients 165 |

control dimension (continued)
|PerfCubes 166 |
|Processes 166 |
|StatsByClient 166 |
|StatsByCube 167 |
|StatsByCubeByClient 168 |
|StatsForServer 168 |
|TimeIntervals 168 |
controlling object security 43
converting .xls 124
converting worksheets 125
copy

|dimension 139 |
|Counter Name 88 |
cube 60, 141

|enabling and disabling logging 34, 59 |
mirror 132
replicating 131, 138
security 45
source 132, 138, 139, 140, 141
specifying replication information 139
synchronizing 131, 141
Cube Dependancy 127
Cube Viewer
page size 120
Cube Viewer, export sheets 121
CubeviewerStringWrap 121

data backup 34, 59
data directory
restricting access 51
Data Reservation
capabilities 5
data, recovery 34, 59
DataAdmin group 48
Date and Time 67
DEBUG
Admin server message severity 56
logging system message severity 63
TM1 Web message severity 122
Dependency in Cubes 127
dimension
local 139
rename 139
security 45
specifying replication information 139
dimension, copy 139
DisableSandboxing 5
disconnecting clients 38
Display user names 77
Do Not Copy Rule 140

editing tm1web_config.xml 109

|element security 45 |

End Time 60
ERROR
Admin server message severity 56
logging system message severity 63
TM1 Web message severity 122
exit TM1 Top 84
export
  maximum sheets 121

F
FATAL logging system message severity 63

G
garbage memory 39
  group
    adding 45, 50
    admin 47, 51
    multiple 51

H
heartbeat interval 19, 21

I
idle
  TM1 Top 83
INFO
  Admin server message severity 56
  logging system message severity 63
  TM1 Web message severity 122
  integrated login 43
  IX-lock 83

K
keyboard shortcuts
  accessibility 143

L
LDAP
  authentication 43
  local dimension 139
  local server 22, 31
  lock modes 83
  log
    backing out records 36, 61
    file 25, 34, 59
    file, removing 36, 61
    file, sample 33, 58
    Tm1s.log file 58
    viewing 35, 60
  log, message 62
  LogAppend 78
  logging
    disabling 34, 59
    sample properties file 64
    system message severity 63
    transactions 33, 58
  logging, enabling 34, 59
  login page configuring 112
  LogPeriod 78
  LogReleaseLineCount 36

M
MaskUserNameInServerTools parameter 77
MaxBackupIndex 66
MaxFileSize 66
memory
  garbage 39
  management 39
  MemorySize 66
message
  clients 37
  log 62
  text 67
  Message Level 67
Message log
  configuring 64
  properties file 64
  severity Levels 63
  TM1 loggers 63
  using 62
metadata 25
mirror cube 137, 138
  definition 132
  dimension 139
  naming 139
  rules 139, 140
  specify information 139
  synchronization 141
MTQ 4
multi-threaded queries 4

N
navigation tree
  views node 120
  new features 1

O
object security 43
Optional Tm1s.cfg parameter
  MTQ 4
out-of-process server 131
override dimension server 131
override dimension defaults 139

P
Parallel Interaction
  best practice 13
  enabling 12
  when and how to use 12
parameters, tm1web_config.xml 109
passwords
  assigning 51
  change 109
  for the ADMIN ID 30
  passwords, admin group 51
  passwords, change 52
  passwords, clearing 52
  passwords, setting 52
  performance counters 86
  performance monitoring 37, 76
  |StatsByClient 155
  |StatsByCube 156
  |StatsForServer 157
  control cubes 76, 154
Persistent feeders 8, 10
Personal Workspace Writeback Mode 5
privilege
  read access 135, 141
  write access 141

R
  R-lock 83
  read access privilege 135, 141
remote server
  Admin host 22
  and replication 132
  application 26
  connecting 30
  diagram 25
  disconnecting 31
  memory management 39
  operations 25
  refreshing servers 30
  replication 132
  source server 132, 138
  starting 28
  target server 132, 138
  Windows service 28
removing log files 36, 61
rename a dimension 139
replicate
  a cube 132, 138
  a dimension 139
Admin Host considerations 135
advantages 131
connection 132, 136
dimension name 139
mirror cube name 139
outcome 140
overview 131
process 138
required access privileges 135
server configurations 132
server relationships 132
setting up 136
source server 132
specifying rule information 140
specifying cube information 139
specifying dimension information 139
subset 140
target server 132
views 140
replication connection
  creating 137
  deleting 138
  modifying 137
  overview 132
rollback TM1 Top 83
rules 139, 140
S
sandbox
  behavior in different clients 11
  configuration parameters 11
  delete sandboxes 11
  files and folders 11
  managing 10
  memory usage 11
Sandbox Capability Assignment 5
sandboxes, capability assignments 5
saving data 25
security
  access privileges 45
  adding groups 50
  adding users 49
  assigning passwords 51
  change password 52
  clearing passwords 52
  cube-level 45
dimension-level 45
element-level 45
group 45, 47, 50, 51
multiple groups 51
password security 51
setting passwords 52
TM1 data 51
TM1 objects 43
Security Admin group 47
server
  Admin 19
  audit log 70
  local 22, 31
  out-of-process 131
  performance 76
  performance monitoring 37, 76
  remote 22, 25, 28, 30, 31, 39, 132, 138
  remote shutdown 38
  remote shutting down 31
  serformance 37
  source 132, 138
  star configuration 132
  start and chain configuration 132
  target 132, 138
server, message log 62
ServerName 78
service automatic start 28
Set Rule to Synchronize 140
setting passwords 52
shut down 63
source cube 132, 138, 139, 140, 141
source server 132, 138
star and chain configuration 132
star configuration 132
Stargate view 39
Starting Time 60
SuppressPleaseSaveDialog 117
synchronization
  automatic 142
  on-demand 131, 141
  over unstable networks 142
  overview 132, 141
  required access privileges 141
T
  target server 132, 138
Thread ID 67
TimeZone 66
TM1
  log file 34, 59
  Logger Name 67
  securing data 51
  transaction log file 25, 33, 58
users and groups security 45
TM1 PerfMon 86, 87