IBM Planning Analytics
Version 2 Release 0

TM1 Operations
Note
Before you use this information and the product it supports, read the information in “Notices” on page 167.
Contents

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

Chapter 1. TM1 System and Feature Configuration........................................................................................................... 1
  Configuration Overview.......................................................................................................................................................... 1
  Improving processing performance with multi-threaded queries........................................................................................... 2
  Capability Assignments......................................................................................................................................................... 3
    Cognos TM1 Application Web Capabilities.......................................................................................................................... 5
    Setting Capabilities............................................................................................................................................................... 5
    Understanding the interaction of assignments in Capability Assignments........................................................................... 5
  Using Persistent Feeders.......................................................................................................................................................... 6
    Saving data with persistent feeders....................................................................................................................................... 6
    Modifying rules for cubes with feeders............................................................................................................................. 6
    Handling corrupted or invalidated feeder files.................................................................................................................. 7
  Configuring and Managing Sandboxes in your TM1 Environment.......................................................................................... 7
    Memory Usage Considerations for Sandboxes and Personal Workspaces.......................................................................... 7
    Managing Files and Folders for Sandboxes.......................................................................................................................... 7
    TM1 Server Configuration Parameters for Sandboxes......................................................................................................... 8
    Understanding Sandbox Differences Among Different TM1 Clients.................................................................................. 8

Chapter 2. Cognos TM1 Admin Server Operations............................................................................................................... 9
  TM1 Admin Server................................................................................................................................................................. 9
  Running the Cognos TM1 Admin Server on a Windows operating system........................................................................... 10
    Running the Cognos TM1 Admin Server as an Application on Windows........................................................................ 10
  Running the Cognos TM1 Admin Server on UNIX................................................................................................................. 11
  Viewing Admin Server Status Report.................................................................................................................................. 11
  Specifying the Location of the Admin Host........................................................................................................................... 11
  Specifying Multiple Admin Hosts........................................................................................................................................ 12

Chapter 3. Remote Cognos TM1 Server Operations.............................................................................................................. 13
  Overview of Remote Server Operations............................................................................................................................... 13
  Setting Up a Remote TM1 Server to Run as an Application................................................................................................. 14
  Setting Up a Remote TM1 Server to Run as a Windows Service............................................................................................ 14
    Installing a TM1 Server to Run as a Windows Service...................................................................................................... 15
    Removing a TM1 Server from Running as a Windows Service........................................................................................... 15
  Starting a Remote TM1 Server............................................................................................................................................... 15
    Starting a TM1 Server Set Up as a Windows Application.................................................................................................. 15
    Starting a TM1 Server Installed as a Windows Service.................................................................................................... 16
    Starting a UNIX TM1 Server.............................................................................................................................................. 16
  Connecting to a Remote Server............................................................................................................................................. 17
    Refreshing the List of Remote Servers................................................................................................................................ 17
    Re-Setting the Admin Host................................................................................................................................................ 17
    Re-Setting Local Server Options...................................................................................................................................... 18
  Disconnecting from a Remote Server.................................................................................................................................. 18
  Shutting Down a Windows TM1 Server................................................................................................................................. 18
    Shutting Down a Windows TM1 Server Running as a Service............................................................................................ 18
  Shutting Down a UNIX TM1 Server...................................................................................................................................... 19
    Shutting Down a UNIX TM1 Server Running in Background Mode.................................................................................... 19
  Logging Transactions............................................................................................................................................................... 20
    Data Backup and Recovery................................................................................................................................................ 20
    Enabling and Disabling Logging......................................................................................................................................... 21
    Viewing the Transaction Log.............................................................................................................................................. 21
# Chapter 7. Cognos TM1 Tools and Utilities

**IBM Cognos Performance Management Hub**

- Opening the Performance Management Hub web page.......................... 81
- Configuring logging................................................................................. 82
- tm1xfer........... 83  
  - tm1xfer syntax................................................................................ 83  
  - Compressing directories with tm1xfer........................................... 84  
  - Decompressing directories with tm1xfer...................................... 85  
  - Notes on tm1xfer............................................................................. 85  
  - Error messages for tm1xfer........................................................... 86  
- odbc_test tool...................................................................................... 87  

---

**Chapter 8. Using the IBM Cognos TM1 Operations Console**

**IBM Cognos TM1 Operations Console workflow**

- Configuring the Cognos TM1 Operations Console............................ 89  
- Starting and logging into the Cognos TM1 Operations Console........... 90  

---

**Starting and logging into the Cognos TM1 Operations Console**

- Capturing Core Dumps for TM1 Server Crash Analysis...................... 74
- Monitoring Server Performance Using Control Cubes......................... 55
- Using TM1 Performance Counters...................................................... 65
- Enabling Performance Monitoring...................................................... 55
- Disabling Performance Monitoring.................................................... 55
- Viewing Performance Statistics for Clients, Cubes, and Servers............ 56

---

**Using the Audit Log**

- Understanding Audit Log Events...................................................... 50
- Configuring Audit Logging............................................................... 51
- Updating the Audit Log with the Latest Events................................. 52
- Using the Audit Log Window to View Log Messages........................... 52

---

**Configuring**

- Client Logging.................................................................................. 49
- Sending email alerts with Logback.................................................... 69
- Specifying the location of the Logback configuration file................ 70  
  - Setting up the Logback configuration file.................................. 70
  - Setting up Logback filters............................................................ 71
  - Sample Logback configuration file.............................................. 74

---

**Capturing Core Dumps for TM1 Server Crash Analysis**

- Using tm1xfer to transmit files to Cognos........................................ 79
- Collecting Files for Analysis Following A TM1 Server Crash.............. 79
- Installing and Running ADPlus......................................................... 75
- Running Dr. Watson............................................................................ 78
- Updating the Audit Log with the Latest Events................................. 52
- Viewing Performance Counters with the Microsoft Windows Performance Monitor.................................................. 68
- Viewing TM1 Performance Counters with the TM1 PerfMon Utility........ 66
- Viewing TM1 Performance Counters with the Microsoft Windows Performance Monitor.................................................. 68
- Important Note about Running TM1 Performance Counter Tools........ 65
- Viewing TM1 Performance Counters.................................................. 65
- Sending email alerts with Logback.................................................... 69
- Specifying the location of the Logback configuration file................ 70
- Setting up the Logback configuration file.................................. 70
- Setting up Logback filters............................................................ 71
- Sample Logback configuration file.............................................. 74

---

**Debugging Tools for Windows**

- Installing and Running ADPlus......................................................... 75
- Running Dr. Watson............................................................................ 78
- Collecting Files for Analysis Following A TM1 Server Crash.............. 79
- Transmitting Files to Cognos............................................................ 79

---

**Troubleshooting:**

- Recovering from a Corrupt Transaction Log File............................... 42
- Using the TM1 Server Message Log................................................ 43
- Message Severity Levels.................................................................. 43
- TM1 Loggers.................................................................................... 44
- Logging Properties File..................................................................... 44
- Configuring and Enabling Server Message Logging......................... 45
- Viewing the TM1 Server Message Log............................................ 48
- Client Logging................................................................................ 49
- Using the Audit Log......................................................................... 50
- Understanding Audit Log Events.................................................... 50
- Configuring Audit Logging............................................................. 51
- Updating the Audit Log with the Latest Events............................... 52
- Using the Audit Log Window to View Log Messages....................... 52
- Monitoring Server Performance Using Control Cubes....................... 55
- Enabling Performance Monitoring.................................................. 55
- Disabling Performance Monitoring................................................ 55
- Viewing Performance Statistics for Clients, Cubes, and Servers........ 56
- Using the TM1 Top Utility................................................................. 56
- Displaying user names in TM1 Top.................................................. 56
- Configuring the Tm1top.ini File....................................................... 56
- Running TM1 Top............................................................................. 58
- Viewing and Understanding the TM1 Top Display............................ 59
- TM1 Top Commands........................................................................ 63
- Canceling a Thread’s Processing...................................................... 64
- Using TM1 Performance Counters.................................................... 65
- Important Note about Running TM1 Performance Counter Tools........ 65
- Available TM1 Performance Counters............................................ 65
- Viewing TM1 Performance Counters with the TM1 PerfMon Utility........ 66
- Viewing Performance Counters with the Microsoft Windows Performance Monitor.................................................. 68
- Sending email alerts with Logback.................................................... 69
- Specifying the location of the Logback configuration file................ 70
- Setting up the Logback configuration file.................................. 70
- Setting up Logback filters............................................................ 71
- Sample Logback configuration file.............................................. 74
- Capturing Core Dumps for TM1 Server Crash Analysis...................... 74
- Debugging Tools for Windows.......................................................... 75
- Installing and Running ADPlus......................................................... 75
- Running Dr. Watson............................................................................ 78
- Collecting Files for Analysis Following A TM1 Server Crash.............. 79
- Transmitting Files to Cognos............................................................ 79

---

**Chapter 7. Cognos TM1 Tools and Utilities**

**IBM Cognos Performance Management Hub**

- Opening the Performance Management Hub web page.................... 81
- Configuring logging................................................................. 82
- tm1xfer....................................................................................... 83
- tm1xfer syntax.............................................................................. 83
- Compressing directories with tm1xfer.......................................... 84
- Decompressing directories with tm1xfer...................................... 85
- Notes on tm1xfer........................................................................... 85
- Error messages for tm1xfer........................................................ 86
- tm1xfer usage notes.................................................................... 87
- odbc_test tool............................................................................. 88
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up the Cognos TM1 Operations Console</td>
<td>90</td>
</tr>
<tr>
<td>Adding an Operation Group, Adminhost, and servers</td>
<td>90</td>
</tr>
<tr>
<td>Verify admin access</td>
<td>91</td>
</tr>
<tr>
<td>Display user names</td>
<td>91</td>
</tr>
<tr>
<td>Server Health Status</td>
<td>91</td>
</tr>
<tr>
<td>Monitoring in the Cognos TM1 Operations Console</td>
<td>92</td>
</tr>
<tr>
<td>Basic and enhanced server status reports</td>
<td>92</td>
</tr>
<tr>
<td>Sandbox and sandbox queue reports</td>
<td>93</td>
</tr>
<tr>
<td>Performance Statistics reports</td>
<td>93</td>
</tr>
<tr>
<td>TurboIntegrator Processes reports</td>
<td>93</td>
</tr>
<tr>
<td>Monitoring Chores</td>
<td>94</td>
</tr>
<tr>
<td>Memory Usage Graph</td>
<td>94</td>
</tr>
<tr>
<td>Thread Details Graph</td>
<td>95</td>
</tr>
<tr>
<td>Log files in the IBM Cognos TM1 Operations Console</td>
<td>99</td>
</tr>
<tr>
<td>Working with logs</td>
<td>99</td>
</tr>
<tr>
<td>TM1 server Transaction log</td>
<td>103</td>
</tr>
<tr>
<td>TM1 server Message log</td>
<td>103</td>
</tr>
<tr>
<td>TM1 server Audit log</td>
<td>103</td>
</tr>
<tr>
<td>Using Watchdog to monitor server activity</td>
<td>104</td>
</tr>
<tr>
<td>Chapter 9. Administering IBM Cognos TM1 Web</td>
<td>107</td>
</tr>
<tr>
<td>IBM Cognos TM1 Web Overview</td>
<td>107</td>
</tr>
<tr>
<td>Changing Your Password in Cognos TM1 Web</td>
<td>107</td>
</tr>
<tr>
<td>Configuring a proxy account for relational data connections</td>
<td>107</td>
</tr>
<tr>
<td>Modifying Cognos TM1 Web Configuration Parameters</td>
<td>108</td>
</tr>
<tr>
<td>Cognos TM1 Web Configuration Parameters</td>
<td>108</td>
</tr>
<tr>
<td>Editing the Cognos TM1 Web configuration file</td>
<td>114</td>
</tr>
<tr>
<td>Configuring the Cognos TM1 Web Login Page using AdminHostName and TM1ServerName parameters</td>
<td>114</td>
</tr>
<tr>
<td>Configuring a Custom Homepage for IBM Cognos TM1 Web</td>
<td>115</td>
</tr>
<tr>
<td>Configuring IBM Cognos TM1 Web Startup and Appearance Settings</td>
<td>119</td>
</tr>
<tr>
<td>Changing the Cube Viewer Page Size</td>
<td>122</td>
</tr>
<tr>
<td>Setting the Maximum Number of Sheets to Export from a Cube Viewer</td>
<td>122</td>
</tr>
<tr>
<td>Wrapping string values in cube views</td>
<td>122</td>
</tr>
<tr>
<td>Using IBM Cognos TM1 Web Logging</td>
<td>123</td>
</tr>
<tr>
<td>IBM Cognos TM1 Web log file</td>
<td>123</td>
</tr>
<tr>
<td>Message Severity Levels for IBM Cognos TM1 Web Logging</td>
<td>123</td>
</tr>
<tr>
<td>Configuring and enabling IBM Cognos TM1 Web logging</td>
<td>124</td>
</tr>
<tr>
<td>Viewing the IBM Cognos TM1 Web Log File</td>
<td>125</td>
</tr>
<tr>
<td>Microsoft Excel .xls worksheets</td>
<td>125</td>
</tr>
<tr>
<td>Converting a .xls worksheet to .xlsx</td>
<td>126</td>
</tr>
<tr>
<td>Checking default font settings for non-Microsoft Windows web servers</td>
<td>126</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>128</td>
</tr>
<tr>
<td>Creating Cube Dependencies</td>
<td>128</td>
</tr>
<tr>
<td>Using AddCubeDependency to Establish Dependencies</td>
<td>129</td>
</tr>
<tr>
<td>Debugging and Identifying Cube Dependencies</td>
<td>129</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>134</td>
</tr>
<tr>
<td>Admin Server Considerations</td>
<td>134</td>
</tr>
<tr>
<td>Configuring the Tm1s.cfg File to Support Replication</td>
<td>135</td>
</tr>
<tr>
<td>Maintaining Replication Connections</td>
<td>135</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 documentation on the web, including all translated documentation, access IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter).

Samples disclaimer

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

Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products. IBM Cognos TM1 has some components that support accessibility features. IBM Cognos TM1 Performance Modeler, IBM Cognos Insight, and Cognos TM1 Operations Console have accessibility features.

See Accessibility 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.

Security considerations

For security considerations for IBM Planning Analytics, see Planning Analytics Installation and Configuration. Information on managing user and group authentication can be found in the Managing Users and Groups chapter of the TM1 Operations documentation.
# Chapter 1. TM1 System and Feature Configuration

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. 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 <em>Planning Analytics Installation and Configuration</em>.</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 <em>Planning Analytics Installation and Configuration</em>.</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.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Improving processing performance with multi-threaded queries” on page 2.</td>
</tr>
<tr>
<td>Capability Assignments</td>
<td>A set of capabilities that administrators can enable or disable by user group. 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 “Capability Assignments” on page 3.</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 “Using Persistent Feeders” on page 6.</td>
</tr>
<tr>
<td>Sandboxes</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>For details, see “Configuring and Managing Sandboxes in your TM1 Environment” on page 7.</td>
</tr>
</tbody>
</table>
**Configuration** | **Description**
--- | ---
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 \CubeProperties 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 IBM Cognos TM1 for Developers.

---

### Improving processing performance with multi-threaded queries

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

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

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

- **MTQ**: Sets the maximum number of threads per TM1 query.
- **MTQ.CTreeRedundancyReducer**: Reduces the probability of recalculating the same rule-driven cells in MTQ worker threads.
- **MTQ.OperationProgressCheckSkipLoopSize**: Specifies the number of cells to be processed before checking whether multi-threaded splits are needed.
- **MTQ.SingleCellConsolidation**: Disallow multi-threaded query processing for single cell consolidations.

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

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

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

You can track multi-threaded query execution using the Enhanced object contention report in Cognos TM1 Operation Console.
Note: TM1 server multi-threaded queries are no longer limited to 64 CPU cores. Microsoft Windows Platforms with more than 64 processors can take full advantage of this feature and its performance benefits.

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

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

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

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

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

The following capabilities can be set per user group:

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

- **Personal Workspace Writeback Mode**
  This capability defines how data changes are handled in this user group.

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

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

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

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

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

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

- **Sandbox**

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

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

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

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

- **Data Reservation capabilities**

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

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

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

- **Consolidation Type In Spreading**

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

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

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

  A blank setting is equivalent to Grant.

- **Allow Spreading**

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

  A blank setting is equivalent to Grant.

- **Allow Export as Text**

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

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

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

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

**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 the default case.</td>
<td>Deny</td>
<td>Grant</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 &quot;default&quot; 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 setting is similar to the addition of DisableSandboxing=T to the configuration file which prevents the use of sandboxes and puts all users into direct writeback.</td>
<td>Deny</td>
<td>Deny</td>
</tr>
</tbody>
</table>

**Understanding the interaction of assignments in Capability Assignments**

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

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

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

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

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

See the “Using Personal Workspaces and Sandboxes” in the *TM1 Architect, Perspectives, and TM1 Web* documentation 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 <em>Personal Workspace</em>.</td>
</tr>
<tr>
<td>Personal Workspace</td>
<td>Commit</td>
<td>Sends the <em>Personal Workspace</em> data to the TM1 Base Model.</td>
</tr>
</tbody>
</table>
### 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 *Planning Analytics Installation and Configuration*.

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` parameter set to T. As part of the server startup the `.feeders` files for all the relevant cubes will be written. A new persistent feeders file is written for any cube for which the persistent feeders file is not found at server startup. This means that the `.feeders` files may be deleted from the data directory to force a complete re-calculation of feeders when the server next starts. In a TI process, use the `DeleteAllPersistentFeeders()` TI function to delete all persistent feeders.

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

#### Modifying rules for cubes with feeders

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

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

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

### Handling corrupted or invalidated feeder files

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

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

```
27660 ERROR 2010-05-14 19:10:26.455 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 *TM1 Architect, Perspectives, and TM1 Web* and *TM1 TurboIntegrator* documentation.

### 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 *Planning Analytics Installation and Configuration*.

### Managing Files and Folders for Sandboxes

Each TM1 server stores the support files for sandboxes in the sub-folders of the TM1 server data directory.

If you are backing up this data, you may want to consider these files and the implications of backing up and restoring sandbox files.
Administrators can delete sandboxes using a variety of criteria.

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

**TM1 Server Configuration Parameters for Sandboxes**

The TM1 server configuration file, Tm1s.cfg, uses parameters for enabling and managing the sandbox feature.

- DisableSandboxing parameter
- MaxUserSandboxSize parameter

For more details about the Tm1s.cfg file and these parameters, see Planning Analytics Installation and Configuration.

- 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 Operations documentation 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 dropdown list.
Chapter 2. 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 Planning Analytics Installation and Configuration.

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 TM1 TurboIntegrator documentation 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 a Windows application. For details, see “Running the Cognos
TM1 Admin Server as an Application on Windows” on page 10.

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 Planning Analytics Installation and Configuration.

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 Planning Analytics Installation and Configuration.

**Viewing Admin Server Status Report**

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

The Admin Server report contains the following information:

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

**Specifying the Location of the Admin Host**

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

- To specify the Admin Host referenced by clients, edit the AdminHost parameter in the Tm1p.ini client configuration file.
  
  You can change the Tm1p.ini file by using the TM1 Options menu in Server Explorer.

  You can also manually edit the Tm1p.ini file. For more information, see the Tm1p.ini topic in Planning Analytics Installation and Configuration.
To specify the Admin Host with which remote servers register, use one of the following methods:

- Edit the AdminHost parameter in the Tm1s.cfg file.
- Use the -v command-line parameter when you bring up the Windows version of the TM1 server.

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

**Specifying Multiple Admin Hosts**

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

A client that specifies multiple Admin Hosts can access any TM1 servers that are registered with the Admin Servers on the specified hosts.
Chapter 3. Remote 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 Planning Analytics Installation and Configuration. For details on using the IBM Cognos Configuration utility to manage TM1 remote servers on a Windows or UNIX system, see Planning Analytics Installation and Configuration.

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 “Backing Out Records from the Transaction Log” on page 22.

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 Tm1syyyymmddhhmmss.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.

Any changes to the metadata, such as dimension definitions and cube definitions, are immediately saved to disk. The changes to the metadata are not written to the transaction log file.

### 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:

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

8. Click OK to save the shortcut.

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

### 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: `tm1sd.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>
</tbody>
</table>
| -u        | Optional          | Valid user name for the computer on which you are working. Use one of the following formats:  
• Domain and user name in the format `Domain\username`. For example, `entp\jsmith`.  
• User name for a local user account.  
If you do not provide a user name, the current user account is used by default. |
| -w        | Optional          | Password for the above account. |

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

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

**Removing a TM1 Server from Running as a Windows Service**

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

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

where `ServerName` is the Microsoft Windows service name of the TM1 server you want to remove.

For example, the following command line removes an existing service for the sdata TM1 server.

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

**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.
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:

> ./startup_tm1s.sh /software/AutoInstall/TM1/samples/tm1/SData
You can save to a new file name if desired. If you have installed multiple TM1 servers, you should create server-specific scripts with unique file names for each server.

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

**Running the UNIX TM1 Server in Background Mode**

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

```
RunningInBackground=T
```

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

See also “Shutting Down a UNIX TM1 Server Running in Background Mode” on page 19.

### 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 23.

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 the shutdown utility, tm1srvstop.exe, located in the install_dir/bin directory.

The shutdown utility tm1srvstop.exe accepts the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>Required 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 IP address of the Admin Server on which the TM1 server is registered.</td>
</tr>
<tr>
<td>-user</td>
<td>Required 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 Password for the person bringing down the server.</td>
</tr>
<tr>
<td>-time</td>
<td>Optional Delay, in seconds, at which the server will be shut down after running shutdown_tm1s.sh</td>
</tr>
<tr>
<td>-cancel</td>
<td>Optional Cancels any previously issued shutdown_tm1s.sh commands. Valid values are T and F.</td>
</tr>
<tr>
<td>-unsave</td>
<td>Optional By default, TM1 saves the data before a TM1 server is shut down with the shutdown_tm1s.sh command. The -unsave parameter shuts down a server without saving the data. Valid values are T and F.</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.

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

Alternatively, you can use shutdown_tm1s.sh to shut down a server. This method has the following limitations, however:
• The shutdown_tm1s.sh script works only on the local machine where TM1 Server is running.
• If you use the startup_tm1s.sh and shutdown_tm1s.sh scripts, you must ensure that the following line exists in the TM1 Server configuration file (tm1s.cfg): RunningInBackground=T.

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","Budget","Belgium","L Series 1.6 L Šedan","Units","Sep","" "19980602212749","19980602212749","Admin","N","610.","800.","salescube","Budget","Belgium","L Series 1.6 L Šedan","Units","Oct",""
```

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:

Tm1syyyymmddhhmms.log

The time stamp, yyyymmddhhmms, 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, 2015 at 2:30 PM GMT, TM1 returns all the records in all the transaction logs with a time stamp of 20150102143000 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.

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 Planning Analytics Installation and Configuration documentation.

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 143.
<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 *Planning Analytics Installation and Configuration*.
   
   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 }StatsForServer control cube, which is described in Appendix B, “Control Cubes,” on page 143.

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

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

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

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

About Stargate Views

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

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

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

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

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

Using TM1 in Bulk Load Mode

Bulk Load Mode enables IBM 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 need 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 UsingBulk 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 connections will also be suspended.

Ending Bulk Load Mode
When Bulk Load Mode is disabled, all system and user threads will be resumed and user logins will be allowed.

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

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

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

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

```
TM1V TM1ServerEnableBulkLoadMode(TM1P hPool, TM1Server hServer)
TM1V TM1ServerDisableBulkLoadMode(TM1P hPool, TM1Server hServer)
```

where:
• `TM1P` is a pool handle obtained with `TM1ValPoolCreate`.
• `TM1Server` is a handle to the current server.

TurboIntegrator Process Commands
You can enable Bulk Load Mode in either the Prolog or Epilog section of a 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 4. 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 *Planning Analytics Installation and Configuration*.

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>
<thead>
<tr>
<th>Table 1: Cognos TM1 server authentication methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authentication Methods</strong></td>
</tr>
<tr>
<td>TM1 Authentication</td>
</tr>
<tr>
<td>Integrated Login</td>
</tr>
<tr>
<td>LDAP Authentication</td>
</tr>
<tr>
<td>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 IBM Cognos *TM1 for Developers*. 

© Copyright IBM Corp. 2007, 2017
Chapter 5. 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.
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 cannot view the data in non-security TM1 objects, such as cubes, dimensions and rules. The members in this group are allowed to create, edit and delete TM1 users and groups. Additionally, these users can manage the access 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 IBM Cognos TM1 for Developers.

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.
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 *Planning Analytics Installation and Configuration*.

**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 groups.

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

**Membership in Multiple Groups**
A user who is a member of multiple groups receives the highest level of 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 6. 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 real-time performance monitoring of the TM1 servers in your organization.

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

- **TM1 Operations Console**
  A web-based utility that displays and logs TM1 server activity. See the *IBM Cognos TM1 Operations* documentation.

  You can use TM1 Operations Console to set up and view the following log files.

  - **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, the name of the client who made the change, before and after value, name of the cube in which the change was made, and the elements that identify the cell that changed.

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

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

- **Monitoring the TM1 Admin server**

  The Admin Server log file contains messages about communication between TM1 clients, the Admin Server and individual TM1 servers.

- **Monitoring Server Performance Using Control Cubes**

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

- **Performance Counters**

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

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 Planning Analytics Installation and Configuration.

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:

```java
# 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:

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

For example, replace INFO with the DEBUG logging level:

```java
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</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</td>
<td>TM1.Event</td>
<td>Destroy event 0043B858 succeeded</td>
</tr>
</tbody>
</table>

System and Performance Monitoring 39
**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:

<table>
<thead>
<tr>
<th>Date and Time</th>
<th>Client Name</th>
<th>Change Type</th>
<th>Old Value</th>
<th>New Value</th>
<th>Cube Name</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>19980602212741</td>
<td>&quot;Admin&quot;</td>
<td>&quot;N&quot;</td>
<td>380.</td>
<td>250.</td>
<td>salescube</td>
<td>19980602212741</td>
<td>&quot;Budget&quot;, &quot;Belgium&quot;, &quot;L Series 1.6 L Sedan&quot;, &quot;Units&quot;, &quot;Jun&quot;, &quot;&quot;</td>
</tr>
<tr>
<td>19980602212744</td>
<td>&quot;Admin&quot;</td>
<td>&quot;N&quot;</td>
<td>430.</td>
<td>600.</td>
<td>salescube</td>
<td>19980602212744</td>
<td>&quot;Budget&quot;, &quot;Belgium&quot;, &quot;L Series 1.6 L Sedan&quot;, &quot;Units&quot;, &quot;Sep&quot;, &quot;&quot;</td>
</tr>
<tr>
<td>19980602212749</td>
<td>&quot;Admin&quot;</td>
<td>&quot;N&quot;</td>
<td>610.</td>
<td>800.</td>
<td>salescube</td>
<td>19980602212749</td>
<td>&quot;Budget&quot;, &quot;Belgium&quot;, &quot;L Series 1.6 L Sedan&quot;, &quot;Units&quot;, &quot;Oct&quot;, &quot;&quot;</td>
</tr>
</tbody>
</table>

**Data Backup and Recovery**

When you bring down the server normally, TM1 renames the log file by appending a time stamp to it, using the following naming convention: Tm1s yyyyymmddhhmss.log.

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

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

### 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 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 41.

A sample logging properties file is provided in the directory of some sample TM1 databases 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:

```plaintext
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=5 MB

#Specify the max filesize
log4j.appender.S1.MaxFileSize=10 MB

#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 46 and “Configuring Logging Output” on page 46.

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.

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

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

```java
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:

```java
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.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.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.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.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 MaxBackupIndex option. |
| 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>
<tbody>
<tr>
<td>Thread ID</td>
<td>ID number of the thread in the TM1 server that generated the logging event. Example: 2488</td>
</tr>
<tr>
<td>Message Level</td>
<td>Severity level of the message being reported: DEBUG, INFO, WARN, ERROR, or FATAL.</td>
</tr>
<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 “Configuring Logging Output” on page 46. Reported in the format: yyyy-mm-dd hh:mm:ss,milliseconds 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.
A client-side logging tool is available that can collect data about activity for TM1 clients, such as IBM Cognos TM1 Architect and IBM Cognos TM1 Perspectives.

Due to the large amounts of data that can be collected, this tool should only be used under the direction of customer support.

Here is a sample client log:

```log4j.rootLogger=DEBUG, R1
#Server Explore
log4j.logger.COrionMainApp=DEBUG
log4j.logger.CCubeView=DEBUG
log4j.logger.CCubeViewGrid=DEBUG
log4j.logger.COrionTreeview=DEBUG
log4j.logger.COrionApp=DEBUG
log4j.logger.CMultiSelect=DEBUG
log4j.logger.CCreateSandbox=DEBUG
log4j.logger.loggerViewDimension=DEBUG
#Subset Editor
log4j.logger.CSubsetWnd=DEBUG
log4j.logger.CSubsetEntry=DEBUG
log4j.logger.CSubsetListProp=DEBUG
log4j.logger.CSubsetListBox2=DEBUG
log4j.logger.CSubsetListView1=DEBUG
log4j.logger.CSubsetRegExp=DEBUG
log4j.logger.CSubsetSelByAttr=DEBUG
log4j.logger.CSingleSelect=DEBUG
#ISB
log4j.logger.CXLCtrlsApp=DEBUG
log4j.logger.CConnectServerDlg=DEBUG
log4j.logger.CViewCtrl=DEBUG
log4j.logger.CGetViewDlg=DEBUG
log4j.logger.CubeViewerISB=DEBUG
log4j.logger.ExcelReportPrintDPage=DEBUG
log4j.logger.CTM1Connector=DEBUG
log4j.logger.ExcelReportPrintIncludedSheetsPage=DEBUG
log4j.logger.CAutoExcel=DEBUG
log4j.logger.SubsetEditorIntLog=DEBUG
#Perspectives
log4j.logger.TM1Perspectives=DEBUG
log4j.logger.TM1AF=DEBUG
log4j.logger.TM1ExcelAPI=DEBUG
log4j.appendender.R1=org.apache.log4j.RollingFileAppender
log4j.appendender.R1.File=c:\\ClientDebugLog.log
log4j.appendender.R1.MaxFileSize=10 MB
log4j.appendender.R1.MaxBackupIndex=50
log4j.appendender.R1.layout=org.apache.log4j.PatternLayout
log4j.appendender.R1.layout.ConversionPattern=%X{pid} %t %p %d [%Y-%m-%d %H:%M:%S,%Q] %c %m%n
log4j.appendender.R1.layout.TimeZone=GMT```

To turn on client-side logging, create a file called `tm1p-log.properties` in the C:\Documents and Settings \Administrator\Application Data\Applix\TM location on the client machine. The TM1p.ini file must be in the same folder.

Be sure that location is also set in the `log4j.appendender.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.appendender.R1.File=c:\\ClientLog.log` Be sure to use `\\` slashes.
If you want to use a temp directory, specify the location using the other slash:

```
log4j.appender.R1.File=c:/temp/ClientLog.log
```

Be sure to update the file parameter with the new location if you change the tm1p-log.properties file, for example:

```
location.log4j.appender.R1.File=c:\<new location>\ClientDebugLog.log
```

If the presence of this file and at least one component within it is set to DEBUG, client-side logging is collected. `log4j.rootLogger=DEBUG` can only be set to OFF or DEBUG (which turns client logging ON).

To turn off logging, remove or rename this file. You can turn off logging for an individual component by changing the value from DEBUG to OFF.

To customize the log file, you can change:

- Location of the log file is set in `log4j.appender.R1.File=c:\\ClientDebugLog.log`
- Log file max size is set in `log4j.appender.R1.MaxFileSize=10 MB`. When this size is met a new log file is created.
- Backup index is set in `log4j.appender.R1.MaxBackupIndex=50`.

### 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 51.

### 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
- Groups
- Processes
- Rules
- Subsets
- Views.
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 *Planning Analytics Installation and Configuration*.

### 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 non-adjacent 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 icon 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.
   The Find dialog box appears.
2. Enter the text you want to find
3. Click Find Next.
   If the text is found, the cell containing the matching text is highlighted in the Results grid.

**Exporting Results**
The Results panel also includes an export option for exporting results in XML, comma delimited, and tab delimited file formats.

The export feature is available in both the Audit Log and Audit Log Details windows.

To export audit log data, complete the following steps.
Procedure

1. Click **Export** in the Results toolbar. The Save As dialog box appears.
2. Enter a filename and location for the file.
3. Select the file format using the **Save as type** option.
   - XML (*.xml)
   - CSV (Comma delimited) (*.csv)
   - Text (Tab delimited) (*.txt)
4. Click **Save**.

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 143.

<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


   **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 56.

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>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</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.</td>
</tr>
<tr>
<td></td>
<td>The format for this parameter is: logperiod=nn</td>
</tr>
<tr>
<td></td>
<td>where nn is the number of elapsed seconds between updates to the log file. Setting this to zero disables logging.</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>decrease the amount of data written to the file.</td>
</tr>
<tr>
<td></td>
<td>The default value is 2 seconds.</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</td>
</tr>
<tr>
<td></td>
<td>is started. If the LogAppend parameter is not set, the existing log file will be overwritten.</td>
</tr>
<tr>
<td></td>
<td>Valid values are T and F as shown in the following examples:</td>
</tr>
<tr>
<td></td>
<td>If you set LogAppend=T, log data is appended to the log file specified by the LogFile parameter.</td>
</tr>
<tr>
<td></td>
<td>If you set LogAppend=F, the existing log file will be over-written.</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>where nn is the time interval in seconds.</td>
</tr>
<tr>
<td></td>
<td>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” on page 59.

**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 56.
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.

```plaintext
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.

```plaintext
tm1top.exe -i path-to-initialization-file
```
Running TM1 Top with Command-line Options
You can over-ride the basic parameter values in the Tm1top.ini file by entering them on the command line when starting TM1 Top.

Note: SSL security parameters can not be entered at the command line and must be specified in the Tm1top.ini file if TM1 is configured to use SSL. For details, see "Configuring the TM1 Top Utility to Use SSL."

Use the following format to run TM1 Top with command-line options:

```
tm1top.exe -OptionName1 OptionValue1  -OptionName2 OptionValue2  ...
```

Where OptionName and OptionValue can be any of the following parameter and value combinations:
- -adminhost admin-host-name
- -servername server-host-name
- -refresh refresh-period
- -logfile file-path
- -logperiod nnn
- -logappend T or F

For example, to run TM1 Top with the ServerName parameter set to sdata and the refresh parameter set to 5 seconds, enter the following:

```
tm1top.exe  -servername sdata  -refresh 5
```

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

```
tm1top.exe  -servername "planning sample"
```

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

```
tm1top.exe  /?
```

For more details on TM1 Top parameters, see “Configuring the Tm1top.ini File” on page 56.

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>ID</td>
<td>The thread ID of the underlying system thread in the TM1 server.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>User Name</td>
<td>The name of the user or process that started the thread. This can be one of three possible thread types.</td>
</tr>
<tr>
<td></td>
<td>• <strong>User Thread</strong> - Name of an actual user that is logged into TM1. A value of ??? means the user is currently in the login stage.</td>
</tr>
<tr>
<td></td>
<td>Note: The MaskUserNameInServerTools parameter determines if user names are displayed or masked out in the TM1 Top console display. For details, see “Displaying user names in TM1 Top” on page 56</td>
</tr>
<tr>
<td></td>
<td>• <strong>Chore Thread</strong> - A chore running on the TM1 server. The name of the chore thread is displayed in the following format:</td>
</tr>
<tr>
<td></td>
<td>C:chore-name</td>
</tr>
<tr>
<td>User Name (continued)</td>
<td>The name of the process running under the chore is displayed in the Function field.</td>
</tr>
<tr>
<td></td>
<td>• <strong>System Thread</strong> - A TM1 system process running on the TM1 server. Displayed in the following format:</td>
</tr>
<tr>
<td></td>
<td>Th:name of system thread</td>
</tr>
<tr>
<td></td>
<td>• Th:Pseudo - The Pseudo thread is used to clean up user-defined consolidation (UDC) objects.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• Th:DynamicConf - This thread dynamically reads and updates parameters in the TM1 server configuration file, tm1s.cfg.</td>
</tr>
<tr>
<td>State</td>
<td>The current processing state of a specific thread. This value can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• Idle</td>
</tr>
<tr>
<td></td>
<td>• Run</td>
</tr>
<tr>
<td></td>
<td>• Commit</td>
</tr>
<tr>
<td></td>
<td>• Rollback</td>
</tr>
<tr>
<td></td>
<td>• Wait</td>
</tr>
<tr>
<td></td>
<td>• Login</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Understanding TM1 Thread Processing States” on page 62.</td>
</tr>
<tr>
<td>Function</td>
<td>The current API function that the thread is executing.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Obj Lock Status</td>
<td>Lock status of the last object that was locked.</td>
</tr>
<tr>
<td></td>
<td>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 61.</td>
</tr>
<tr>
<td></td>
<td>• # is the number of locks for each specific lock mode.</td>
</tr>
<tr>
<td></td>
<td>Note: The name of the object is displayed in the State field.</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
User Lock Status | Total lock status and count for all objects being used by a thread. Displayed in the format: \( R# \) IX# \( W# \)
Where:
- \( R \), IX, and \( W \) represent the different lock modes on the objects. For more information, see “Understanding TM1 Lock Modes” on page 61.
- \( \# \) is the total number of locks, across all objects, for each specific lock mode.

| Time | The total time, in seconds, that the current API function or chore process has been processing. The name of the API function or chore process is displayed in the Function field. |

#### Sandbox display mode
You can switch TM1 Top to sandbox mode.

Pressing the s key while TM1 Top is running switches to sandbox mode. Pressing n returns you to the normal TM1 Top display.

In addition to the ID and Username fields, the following fields become available in sandbox mode:

| Fields          | Description |
--- | ---|
Active Sandbox | Name of the currently active sandbox. |
Total Sandbox Memory | The Total Sandbox Memory being consumed for all sandboxes by user. |
Number of Sandboxes | Number of sandboxes for this user. |

#### Queue display mode
You can switch TM1 Top to queue mode.

Pressing the q key while TM1 Top is running switches to queue mode. Pressing n returns you to the normal TM1 Top display.

In addition to the ID and Username fields, the following fields become available in queue mode:

| Fields          | Description |
--- | ---|
Name | Name of the sandbox in the queue. |
Node Name | Name of the node for the sandbox in the queue. |
Status | Date, time, and state when the status was taken for the sandbox in the queue. |
Time in Queue | Length of time sandbox has been in the queue before it was processed. |

#### Understanding TM1 Lock Modes
TM1 uses a set of three lock modes to control access to TM1 data.

When the TM1 server is running, these locks are applied to individual objects, such as cubes, views, and dimensions, as these objects are accessed.

TM1 Top displays the status of these locks for the objects being used by the current threads running in a TM1 server. Lock status is displayed by TM1 Top under the State, Obj Lock Status, and Total Lock Status fields.

The lock modes for TM1 objects are described in the following table.
### Understanding TM1 Lock Modes

<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>
| 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 61.  
• 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. |
<p>| 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. |</p>
<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait</td>
<td>Thread is currently waiting for a specific condition to complete. Displayed in the format: \n  \n  Wait: \n  \n  Wait Status - (Object Name) - Object Type \n  \n  Where: \n  \n  • Wait Status can be one of the following conditions: \n  \n  – WR - (WaitForWriterEvent) Thread is waiting for R-locks to be released so it can obtain a W-lock on the object. \n  \n  – 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. \n  \n  – 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. \n  \n  – 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. \n  \n  – WC - (WaitForCompletionEvent) Thread is waiting for another thread to complete and release its locks. \n  \n  – DRR - (Data Reservation Release) Thread is waiting for a Data Reservation to be released. \n  \n  • Object Name is the name of the last TM1 object that was being acted on by the API function displayed in the Function field. \n  \n  • Object Type is a numeric value representing the type of TM1 object.</td>
</tr>
</tbody>
</table>

| 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:

<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. \n  \n  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. \n  \n  <strong>Note:</strong> 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>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</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>
<tr>
<td>Up, Down arrow keys</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Canceling a Thread's Processing**
A system administrator can use TM1 Top to request the cancellation of certain thread operations or functions that are processing-intensive.

TM1 Top can cancel only threads with a state of **Run** that are running functions involving TurboIntegrator processing, constructing a view, or calculating a view.

The state of a thread is indicated under the **State** field in the TM1 Top display.

This functionality provides a way to cancel an action that is negatively impacting other threads by consuming too much of the server's processing time. For example, a TI process that gets stuck in a loop, or when a user tries to compute a very large and demanding view.

To cancel a thread, you must use the **Cancel** and **Verify** commands together.

**Understanding the Cancel Process**
The Cancel process cancels a running thread.

When a thread is in a Run state, and the process involves TurboIntegrator processing, constructing a view, or calculating a view, then the thread can be 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.

**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
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
```

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

```
tm1perfmon -servername salesdata
```

**Note:** You can turn updates to performance counters on or off for a TM1 server by setting PerfMonIsActive in the tm1s.cfg file. See PerfMonIsActive.

Viewing the TM1 PerfMon Display Output

You can view display output from TM1 PerfMon.
C:\>tm1perfmon -server cdata

TM1 Version: 9.0.3.226

Output data and time Tue Dec 05 17:36:42 2006

<table>
<thead>
<tr>
<th>Counter Name/ Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threads: Threads created</td>
<td>6</td>
</tr>
<tr>
<td>Threads: Threads Creation Retry Attempts</td>
<td>0</td>
</tr>
<tr>
<td>Threads: Thread Creations Failures</td>
<td>0</td>
</tr>
<tr>
<td>Views: Views Created</td>
<td>23</td>
</tr>
<tr>
<td>Views: Empty Views Created</td>
<td>0</td>
</tr>
<tr>
<td>Views: Views Destroyed</td>
<td>0</td>
</tr>
<tr>
<td>Dimensions: Dimensions Created</td>
<td>0</td>
</tr>
<tr>
<td>Dimensions: Dimensions Destroyed</td>
<td>119</td>
</tr>
<tr>
<td>Cubes: Cubes Invalidated</td>
<td>0</td>
</tr>
<tr>
<td>Cubes: Dependent Cubes Invalidated</td>
<td>1</td>
</tr>
<tr>
<td>Subsets: Subset Create Empty</td>
<td>0</td>
</tr>
<tr>
<td>Subsets: Subset Calculated by Expression</td>
<td>0</td>
</tr>
<tr>
<td>Subsets: Subset Calculated with Empty Expression</td>
<td>0</td>
</tr>
<tr>
<td>Subsets: Subset Calculated by Expression</td>
<td>0</td>
</tr>
<tr>
<td>Subsets: Subset Created by Expression</td>
<td>0</td>
</tr>
<tr>
<td>Subsets: Subset Created by MDX Expression</td>
<td>0</td>
</tr>
<tr>
<td>Subsets: Subset Duplicated</td>
<td>0</td>
</tr>
<tr>
<td>Subsets: Subsets destroyed</td>
<td>0</td>
</tr>
<tr>
<td>Subsets: Subsets Deleted</td>
<td>0</td>
</tr>
</tbody>
</table>

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

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

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

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

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

**Opening the Microsoft Windows Performance Monitor**
Open the Microsoft Windows Performance Monitor.

**Procedure**
Click **Start, Settings, Control Panel, Administrative Tools, Performance.**
The Microsoft Windows Performance Monitor opens.

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

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

**Procedure**
1. Click **Add** in the Performance Monitor toolbar.
   The Add Counters dialog box displays.
   **Note:** Microsoft Windows can sometimes display the following message when you open or close the Add Counters dialog.
   "At least one data sample is missing. Data collection is taking longer than expected. You might avoid this message by increasing the sample interval. This message will not be shown again during this session."
   This message is informational only and typical if the system becomes too busy when adding new counters. Click **OK** to clear the message.
2. Select **IBM Cognos TM1** from the Performance object list.
   The available performance counters display in the performance counter list.
3. Select a performance counter from the list of available counters.
   You can also click Explain to see a text description of the currently selected counter.
4. Select the instance of the server you want to monitor.
5. Click **Add** to add the selected counter to the performance display.
6. Click **Close** to close the Add Counters dialog and return to the Performance Monitor.
The Performance Monitor will add the TM1 performance counters to the monitoring process and update the display as shown in the following diagram.

Sending email alerts with Logback

You can send email alerts based on server and process activity using Logback. For example, you can send an email alert when an IBM Cognos TM1 server goes offline or when the number of threads reaches a threshold. You can also send email alerts for Watchdog events that are specified in TM1 Operations Console.

Before you perform these steps, you must be familiar with the Logback utility. Logback is an open source utility, not an IBM product. For information about this utility, search the Internet for Logback documentation. For example, see http://logback.qos.ch/manual/index.html.

You can log events and generate email alerts in two ways:

1. You can generate email alerts for server activity. When a server goes online or offline, TM1 automatically generates an event. You can decide whether you want to email alerts or not for these events using filters in the Logback configuration file. You can also set up email alerts for other server activity, such as when the number of threads reaches a threshold.

   To generate these types of alerts, follow these steps:
   a. Specify the location of the Logback configuration file
   b. Set up the Logback configuration file
   c. Set up filters

2. You can generate emails for events that are logged by Watchdog. In this case, you create Watchdog rules to specify what events to capture, and then use the Logback configuration file to set up email alerts. For example, you can create a Watchdog rule to log an event when a certain process runs, and then configure Logback to send email alerts when Watchdog events occur.

   To generate these types of alerts, follow these steps:
   a. Specify the location of the Logback configuration file
   b. Set up Watchdog rules in TM1 Operations Console
   c. Set up the Logback configuration file
Specifying the location of the Logback configuration file

Before you can use Logback to send email alerts, you need to configure your web application server with the location of the Logback configuration file, logback.xml.

Before you begin
Stop the TM1 Application Server.

About this task
These steps describe how to configure the location of the Logback configuration file for an Apache Tomcat web server.

Procedure
1. Start the Tomcat monitor:
   a) Open a command prompt.
   b) Go to the directory where Tomcat is installed.
      For example: c:\Program Files\IBM\Cognos\tm1_64\tomcat\bin.
   c) Run the following command: tomcat6w //ES//pmpsvc
2. On the Java™ tab, under Java Options, set Dlogback.configurationFile to [full_path]\logback.xml.
   For example, C:\logs\logback.xml.
3. Restart Tomcat. On the General tab, click Stop. After the Tomcat service has stopped, click Start.
4. Click OK.
5. If you are logging events based on Watchdog rules, restart Watchdog for the servers you want to monitor.

What to do next
The next step is to set up the Logback configuration file.

Setting up the Logback configuration file
Use the Logback configuration file to set up email alerts.

Before you begin
Before you do these steps, you must be familiar with the Logback utility. Logback is an open source utility, not an IBM product. For information about this utility, search the Internet for Logback documentation. The steps in this section provide an overview.

Ensure that you have set the location of the Logback configuration file. See “Specifying the location of the Logback configuration file” on page 70.

About this task
You can set two main types of alert:

• Server activity alerts:
  – Status alerts: send an alert when the status of a server changes, for example when a server goes offline. You specify what servers and statuses to monitor using a TM1StatusFilter filter.
  – Threshold alerts: send an alert when the number of threads, wait threads, or memory that is used reaches a threshold. You specify thresholds using a TM1ThresholdFilter filter.
• Watchdog alerts: send an email when a Watchdog event occurs

You set up alerts by configuring the logback.xml file. TM1 provides a sample file, which you can use as a starting point. The sample file is found in tm1_installation_location\bin64\opsconsoledata\logback-sample.xml.

Note: You must edit the logback-sample.xml file. It will not work as-is.
Alert emails can contain information such as the message generated by Watchdog, a marker with a value such as NOTIFY_ADMIN, and an MDC value that is used to filter the output message. For TM1, you can use the following MDC keys:

```plaintext
MDCKEY_TM1SERVER = "tm1server"
MDCKEY_ADMINHOST = "adminhost"
MDCKEY_CURRENT = "current"
MDCKEY_LAST = "last"
MDCKEY_ALERTTYPE = "alert_type"
```

Alert type can be one of the following values:

```plaintext
ALERTTYPE_STATUS = "status"
ALERTTYPE_THREADS = "threads"
ALERTTYPE_WAITTHREADS = "wait_threads"
ALERTTYPE_MEMORYUSED = "memory_used"
```

**Procedure**

1. Open the sample logback.xml file: `tm1_installation_location\tm1_64\bin64\opsconsoledata\logback-sample.xml`.
2. Save the file with the name logback.xml in the location you specified in the web application server configuration, for example `c:\logs\logback.xml`.
3. Configure the FILE appender. In the `<File>` tag, specify the absolute path to the file you want to use for logging. For example `c:\logs\sample-log.txt`. This is the file that logs will be written to.
4. Configure the EMAIL appender. This appender configures how email alerts are sent.
   a) Set the evaluator class: `<evaluator class="com.ibm.tm1.logback.filters.PassEvaluator">`.
   By default, Logback is configured to send emails only for the most serious error messages. Using the PassEvaluator configures Logback to send emails for less serious messages, such as memory usage, number of threads, and so on.
   b) Configure the email options for your environment. For details, search the Internet for Logback documentation.
   c) Configure the content to display in alert emails using the `<layout>` tag.
   You can use MDC keys.
   For example, you can include the name of the TM1 server.
   ```xml
   <pattern>%date - %message%n - %marker - %mdc{tm1server}</pattern>
   ```
5. If you want to send email alerts for server activity, specify the conditions that will trigger email alerts by configuring the ALERT appender. See “Setting up Logback filters” on page 71.
6. If you want to send email alerts for Watchdog events, configure the WATCHDOGTRAIL appender.
   a) In the `<File>` tag, specify the absolute path to the file you want to use for logging Watchdog events. For example `c:\Program Files\IBM\cognos\tm1_64\bin64\WatchdogActionTrail.log`.
   b) Specify a `<logger>` for Watchdog events. For more details about the `<logger>` tag, search the Internet for Logback documentation.
7. In the `<root>` tag, specify the logging level, DEBUG or ERROR, and specify an `<appender-ref>`. For more details about the `<appender-ref>` tag, search the Internet for Logback documentation.

**What to do next**

If you want to set up email alerts for server activity, the next step is to set up filters. If you are setting up email alerts for Watchdog events, filters do not apply.

**Setting up Logback filters**

You can specify the conditions under which email alerts will be sent by using filters. Use filters when you are setting up email alerts for server activity. Filters do not apply to Watchdog events.

It is a good practice to log to disk first, and then send email alerts when the filters are working in the way you need.
TM1 provides the following filters:

- **TM1StatusFilter**: Use this filter to set up server status alerts.
- **TM1ThresholdFilter**: Use this filter to set up threshold alerts.

For both status and threshold alerts, you can specify the servers and the admin host you want to monitor.

```xml
<tm1server>
    This tag is optional. Specify the TM1 servers for which you want to receive email alerts. You can specify a list of TM1 servers. To receive email alerts for all TM1 servers, skip this tag.

    For example:
    <tm1server>Planning Sample</tm1server>
    <tm1server>SData</tm1server>
</tm1server>

<adminhost>
    This tag is optional. Set the <adminhost> if you want to monitor all the servers on a particular admin host. Or, set it when you have servers with identical names on two different admin hosts but want to monitor only one of them. Skip this tag if you want to monitor servers on all admin hosts.

    For example:
    <adminhost>adminhost 1</adminhost>
    <adminhost>adminhost 2</adminhost>
</adminhost>

**TM1StatusFilter**

The TM1StatusFilter sets the status values to capture or suppress.

```xml
<status>
    For a server, the valid values for <status> are as follows:

    - Online
    - Running
    - Starting
    - Entering Bulk Load Mode
    - Bulk Load Mode
    - Offline
    - Stopping

    For a process, valid values include run, idle, busy, and so on.

    The <status> tag can be in a list:

    <status>running</status>
    <status>offline</status>
</status>

**<OnMatch> and <OnMisMatch>**

Use <OnMatch> and <OnMisMatch> to filter on the servers for which you want alerts to be sent. For example, you can set up a filter that sends alerts for the Planning Sample server only.

You can chain filters together. For example, you can create a filter that sends alerts for all status changes on the Planning Sample server and no alerts for other servers.

<OnMatch> and <OnMisMatch> both take the values ACCEPT, DENY, and NEUTRAL.

- **ACCEPT**: a log is created and processing does not pass to the next filter.
- **DENY**: a log is not created and processing does not pass to the next filter.
- **NEUTRAL**: it keeps going down the chain. If all filters return NEUTRAL, it is logged.
In the sample Logback file, the last filter in the ALERT appender is DenyFilter. This filter always returns DENY. You can attach DenyFilter to the end of the list to make sure that, if all the other filters return NEUTRAL or if there is any doubt, you can get to the end of the filter list without logging anything.

```xml
<filter class="com.ibm.tm1.logback.filters.DenyFilter"/>
```

This example logs all status changes for the server Planning Sample and no status changes for all other servers.

```xml
<filter class="com.ibm.tm1.logging.TM1StatusFilter">
  <tm1server>Planning Sample</tm1server>
  <OnMatch>ACCEPT</OnMatch>
  <OnMismatch>DENY</OnMismatch>
</filter>
```

This example prevents logging for all servers when the status is running.

```xml
<filter class="com.ibm.tm1.logging.TM1StatusFilter">
  <status>running</status>
  <OnMatch>DENY</OnMatch>
</filter>
```

**TM1ThresholdFilter**

TM1ThresholdFilter handles the threads, wait_threads, and memory_used alert types.

```xml
<alerttype>
  This tag specifies the type of event to which the filter applies. The allowed values are:
  • threads
  • wait_threads
  • memory_used
  You can have more than one alert type in a filter.
</alerttype>

<threshold>
  This tag sets values for thresholds. Thresholds must be numeric. For the alert type memory_used, the value is in MB. Thresholds can be in a list, so you can have more than one threshold in a filter.
</threshold>

<direction>
  The <direction> tag is optional. By default, an alert is created on both the up and down directions.
</direction>
```

This example logs messages when the memory usage for any server crosses the 256 MB boundary either up or down.

```xml
<filter class="com.ibm.tm1.logging.TM1ThresholdFilter">
  <alerttype>memory_usage</alerttype>
  <threshold>256</threshold>
</filter>
```
This example logs messages for the Planning Sample and SData servers when threads cross the 50 when going up and then again when they reach 100 and 150.

```
<filter class="com.ibm.tm1.logging.TM1ThresholdFilter">
  <tm1server>Planning Sample</tm1server>
  <tm1server>SData</tm1server>
  <alerttype>threads</alerttype>
  <threshold>50</threshold>
  <threshold>100</threshold>
  <threshold>150</threshold>
  <direction>up</direction>
</filter>
```

### Combining filters

This example causes alerts to be sent when the threads or wait threads go above 50 or below 40 for any server. Alerts will also be sent when the Planning Sample server changes status to anything except *running*.

```
<filter class="com.ibm.tm1.logging.TM1ThresholdFilter">
  <alerttype>threads</alerttype>
  <alerttype>wait_threads</alerttype>
  <threshold>50</threshold>
  <direction>up</direction>
</filter>
<filter class="com.ibm.tm1.logging.TM1ThresholdFilter">
  <alerttype>threads</alerttype>
  <alerttype>wait_threads</alerttype>
  <threshold>40</threshold>
  <direction>down</direction>
</filter>
<filter class="com.ibm.tm1.logging.TM1StatusFilter">
  <status>running</status>
  <OnMatch>DENY</OnMatch>
</filter>
<filter class="com.ibm.tm1.logging.TM1StatusFilter">
  <tm1server>Planning Sample</tm1server>
  <OnMatch>DENY</OnMatch>
</filter>
<filter class="com.ibm.tm1.logging.DenyFilter">
</filter>
```

### Sample Logback configuration file

A sample Logback configuration file is stored in `tm1_installation_location\tm1_64\bin64\opsconsoledata`. You must edit this file to use email alerts. It will not work as shipped.

**Note:** In the sample file, the root logging level is set to DEBUG. This can cause the log file size to grow very rapidly. If you do not need debug logging, set the root logging level to ERROR.

```
<root>
  <level value="ERROR" />
  <appender-ref ref="FILE" />
</root>
```

### 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 debug your server.
Running ADPlus Remotely from a TM1 Client
You can run ADPlus on a different computer.

In some environments, you might want to manage and run ADPlus from the computer on which a TM1 client is running, rather than directly from the TM1 server.

Installing and Running ADPlus on the Server
You can install and run ADPlus on the computer where the TM1 server is running.

Procedure
1. Install the Microsoft Debugging Tools for Windows package.
2. Create a batch file named `Remoteshell.cmd` in the debugging tools installation directory (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/40b9beb1-3578-48f9-93e1-7ca6760c1c151033.mspx.
5. Open the Windows Task Manager and note the process ID (PID) for Tm1s.exe (if running the TM1 server as an application) or Tm1sd.exe (if running the TM1 server as a service). You will need to know the PID when running ADPlus from the TM1 client.

Running ADPlus from a TM1 Client
You can run ADPlus from a computer on which a TM1 client is installed.

Procedure
1. Copy the remote.exe and breakin.exe files from the debugging tools installation directory on the TM1 server computer. (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 yyyymmddhhmms GMT) at which the file was generated and process name is the name of the TurboIntegrator process that caused the errors.
  
  There may be multiple TM1ProcessError.log files associated with the server session that crashed. All TM1ProcessError.log files are stored in the server data directory.

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

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 7. 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>Cognos TM1 Operations Console</td>
<td>Web-based, enhanced version of TM1Top with added features and a graphical user interface. See “IBM Cognos TM1 Operations Console workflow” on page 89.</td>
</tr>
<tr>
<td>Performance Management Hub</td>
<td>A server framework that provides common application services to IBM Cognos Analysis for Microsoft Excel, Operations Console, and other TM1 client applications. You can set configuration options using the web interface. For more information, see “IBM Cognos Performance Management Hub” on page 81.</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 documentation.</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()&quot; in IBM Cognos TM1 TurboIntegrator.</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 “tm1xfer” on page 83.</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 88.</td>
</tr>
<tr>
<td>Debugging tools</td>
<td>See “Debugging Tools for Windows” on page 75 for tools designed specifically for debugging Cognos TM1.</td>
</tr>
</tbody>
</table>

**IBM Cognos Performance Management Hub**

IBM Cognos Performance Management Hub (PMHub) is a common server framework under which IBM Cognos performance management products interoperate. PMHub provides common application services to IBM Cognos Analysis for Microsoft Excel, Operations Console, and other TM1 client applications.

*Figure 1: IBM Cognos Performance Management Hub*

PMHub provides a web-based user interface, which you can use to set configuration options.
Opening the Performance Management Hub web page

You can set configuration options by using the IBM Cognos Performance Management Hub web page.

Procedure

1. In a web browser, go to the following URL: http://[server_name]:9510/pmhub/pm/admin
   
   For example, if you are currently logged into the computer where the TM1 Application Server is running, go to http://localhost:9510/pmhub/pm/admin.
   
   The Administration and Configuration page is displayed.

2. Expand configurations and select a category.
   
   The configuration settings are displayed.

3. To change a value, double-click a cell in the Value column, type a value, and then press Enter.

Configuring logging

You can adjust the default message logging settings for IBM Cognos Performance Management Hub. You can also enable additional logs.

Performance Management Hub (PMHub) logs messages through the WebSphere® Liberty Profile, according to the setting defined in the server.xml file. The server.xml file is located in [installation location]\wlp\usr\servers\tm1\.

By default, all ERROR and WARNING log messages are written to ../logs/tm1_messages.log. The tm1_messages.log file contains all messages, except trace messages, that are written or captured by the logging component. When the file size reaches 20 MB, the file rolls over to ../logs/tm1_messages_{datestamp}.log. Only two files are kept.

The following attributes of the <logging> element define the default log settings in the server.xml file:

```
consoleLogLevel="WARNING"
logDirectory="${wlp.user.dir}/../../*/logs"
messageFileName="tm1_messages.log"
maxFiles="2"
maxFileSize="20"
```

You can adjust these settings, for example, to change the number of log files that are kept, change the value of maxFiles.

You can also enable additional logging.

**console.log**

Contains redirected standard output and standard error messages from the JVM process. The default log level is AUDIT.

**trace.log**

Contains all messages that are written or captured by the product.

For example, to produce advanced trace logging at the "finest" level, for all components that start with "com.ibm.ba," add the following attributes to the logging element in the server.xml file:

```
traceFileName="pmhub_trace.log"
traceFormat="ADVANCED"
traceSpecification="*=audit:com.ibm.ba.*=finest"
```

The attributes of the element are then as follows:

```
consoleLogLevel="WARNING"
logDirectory="${wlp.user.dir}/../../*/logs"
messageFileName="tm1_messages.log"
maxFiles="2"
maxFileSize="20"
traceFileName="pmhub_trace.log"
```
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>
<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.
Volume Serial Number 4C9A-1CE1
Directory of C:\Users\obaluch\Documents\dbfile
05/11/2011 05:54 PM  <DIR>             .
05/11/2011 05:54 PM  <DIR>             ..
05/11/2011 06:11 PM  <DIR>             sales_planning
 0 Files(s)    0 bytes(s)
 3 Dir(s)   91,081,957,376 bytes free
```

To move the database files to a AIX® server, invoke the following command at the command prompt on the Windows platform:

```
C:\Users\obaluch>tm1_xfer sales_planning
```

This action results in the following files:

```
C:\Users\obaluch\Documents\dbfile>dir
Volume in drive C has no label.
Volume Serial Number 4C9A-1CE1
Directory of C:\Users\obaluch\Documents\dbfile
05/11/2011 05:54 PM  <DIR>             .
05/11/2011 05:54 PM  <DIR>            ..
05/11/2011 06:24 PM  <DIR>             sales_planning
 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+xxxx, where xxxx is the UTF-16 code point for that character.

The tm1xfer tool does not automatically modify the tm1s.cfg file. You must manually change any settings such as database path. The tm1xfer tool does not automatically modify the server name, for example the server name inside Microsoft Excel workbook applications.

Decompression Example

On an AIX system, the zipped file called sales_planning.zip is copied over to location on the filesystem where the database directory should be located. To extract the database files, run the following command on AIX:

```
bash-3.2$ ls
aix64 tests sales_planning.zip
```

```
bash-3.2$ tm1xfer sales_planning.zip
```

After:

```
bash-3.2$ ls
aix64 tests sales_planning sales_planning.zip
```

The tm1xfer tool recognizes that sales_planning.zip is a zipped file, and proceeds to unzip the database files and to modify the filenames and files to handle mixed case objects and Unicode encoding. This is done based on the current platform and locale.

Notes on tm1xfer

Keep in mind the following other considerations when using tm1xfer.

The tm1xfer tool compresses and decompresses most files located in the database folder, except the following types: raw stores and dimension differences.

Important: If raw stores are present, run the "Process Audit Log Events..." command for that TM1 server.

Cases are not preserved in pre-Unicode files

When pre-Unicode files exist in the database, the tm1xfer log displays Pre-Unicode file format detected in <filename>. To fix this situation, resave the view, subset, and dimension on in Microsoft Windows operating system then run the tm1xfer tool again. Then transfer the files to UNIX and run the tm1xfer tool again to unzip the database.

Archives created using another application than the tm1xfer tool

The tm1xfer tool can be used to pack and move database files between two TM1 servers running on the same type of platform, for example from Windows to Windows. In this case, the files are compressed in the tm1xfer format, and then decompressed for Windows and the current locale. It is important to remember that tm1xfer filters out a few file types. These include raw stores and dimension differences, so these will be lost.

tm1xfer archives files in a format different from other compression utilities such as WinZip or WinRar. For this reason, archives created using tm1xfer cannot be decompressed using other compression utilities, and vice versa.

If a complete copy of the database files has to be moved between two systems on the same platform type and the same locale, it may be a better idea to use a compression utility that doesn't do any file processing, such as 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>Table 4: Error Messages for tm1xfer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message</strong></td>
</tr>
<tr>
<td>This locale is not supported by your Java installation</td>
</tr>
<tr>
<td>&lt;arg&gt; does not exist</td>
</tr>
<tr>
<td>Cannot resolve &lt;file&gt;: &lt;message&gt;</td>
</tr>
<tr>
<td>Cannot resolve &lt;base_dir&gt;: &lt;message&gt;</td>
</tr>
<tr>
<td>Cannot serialize out list of files: &lt;message&gt;</td>
</tr>
<tr>
<td>Cannot open &lt;filename&gt; for reading: &lt;message&gt;</td>
</tr>
<tr>
<td>Error writing to zip file: &lt;message&gt;</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 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>
<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 ApplicationEntries.dim in the database.</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:

config_dir:
path-to-config-directory:
dataset_name:
user_name:
user_passwd:
sql_query:

The sql_query: line and all subsequent lines are concatenated to make the SQL query.
Chapter 8. Using the IBM Cognos TM1 Operations Console

This section describes how to use the TM1 Operations Console.

IBM Cognos TM1 Operations Console workflow

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

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

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

1. Create your logical operations group.
2. Verify admin access on that server.
3. For security purposes, you can “Display user names” on page 91.
4. The Health Status Check shows the current status of the server at a glance with icons. See “Monitoring in the Cognos TM1 Operations Console” on page 92 for the list.
5. To use the default set of server configuration parameters, skip to the next step. If you want to change the basic parameters, see Configure.
6. To get a quick status report, double-click the server name to create a new Health Status tab as described in “Server Health Status” on page 91. The Health Status tab is populated with the Enhanced version of the status report. You can add tabs and populate those tabs with the other kinds of log files.
7. Click the Configuration mode icon so you can configure other log files.
8. Click the server selection check box. This action reveals the Schedule new log buttons. You can drag the bottom pane lower to reveal the full contents of the window. Click the Schedule new log button and define the parameters for this new log.
9. To see the log, switch to Monitor mode and right-click the server and select View Log. This action opens a new Monitors tab and populates it with the scheduled logs. Nothing displays if the current time is earlier than the start time. You can verify admin access as well. You can Filter the results to make it more readable.
10. After set up, you can save and reload a log file configuration.
11. You can change the layout to suit your needs. You can also repopulate the bottom pane with other log information.
12. Set a Watchdog.
13. You can also monitor the TM1 Applications server in the TM1 Operations Console, as described in “Monitoring the Cognos TM1 Application Server” on page 96.

Configuring the Cognos TM1 Operations Console

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

Procedure

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

   If you want to use Microsoft Internet Explorer 8 with the Cognos TM1 Operations Console see the Microsoft Internet Explorer documentation and ensure these settings:
   - Enable the option to refresh web pages with every visit
   - Disable the options for script debugging
Starting and logging into the Cognos TM1 Operations Console

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

Procedure

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

   where
   - server_name is the computer where the Cognos TM1 Operations Console and your web application server are installed. You can use the keyword localhost if you are currently logged on to the Web server that is running Cognos TM1 Application Web. Or you can use the machine name, domain name, or IP address of the Web server hosting the application.
   - port_number is the port number where your web application server is running. For the version of WebSphere Liberty that is provided with the Cognos TM1 installation, the default port number is 9510. For Apache Tomcat, open the Apache Tomcat server.xml file in the C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\location to determine the port setting that your version of Tomcat is using.

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

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

   See Setting up the Cognos TM1 Operations Console to get started.

Setting up the Cognos TM1 Operations Console

IBM Cognos TM1 TM1 Operations Console setup tasks.

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

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

Adding an Operation Group, Adminhost, and servers

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

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

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

See Monitoring the Cognos TM1 Applications Server for details on how to monitor a TM1 Application server in the TM1 Operations Console.
Verify admin access

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

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

Display user names

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

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

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

Server Health Status

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

**Note:** You must verify admin access before data can display.

<table>
<thead>
<tr>
<th>Table 5: Cognos TM1 Operations Console Health Status icons</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Icon</strong></td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
</tr>
</tbody>
</table>

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

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

If the report contains a refresh rate, the data is updated at the refresh rate specified for that report.
Monitoring in the Cognos TM1 Operations Console

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

To see these reports, right-click the server you want to use, and select Monitor and the report you want. Before data can display, you may need to verify admin access or schedule a log if one is not being automatically scheduled. You can also choose to save the log to disk or export it. See Filtering for details on how to reduce the displays. For some of these reports, drag the lower pane down to reveal the full content of the window.

**Status**
- Basic or Enhanced

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

**Sandbox**
- Shows the threads being used in sandboxes on the server.

**Sandbox Queue**
- Shows the jobs in the sandbox queue.

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

**Processes**
- Shows the thread involved in any TurboIntegrator processes.

**Chores**
- Shows server activity involved with Chores.

**Memory Usage Graph**
- Provides a graphical display of memory usage on the server.

**Thread Details Graph**
- Provides a graphical display of thread activity on the server.

You can use these logs:

**Transaction log file**
- Displays the transactions recorded in Tm1s.log when a TM1 client changes a cube value.

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

**Audit log**
- Displays changes to metadata, such as modifications to dimensions, views and subsets.

### Basic and enhanced server status reports

You can view basic and enhanced versions of the status information for a server.

**About this task**

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

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

**Results**

The report displays in a new Monitor tab. To reduce the content, use the **Filter** button (see “Filtering results in the IBM Cognos TM1 Operations Console” on page 101). To export the log to a file, select **Log to Disk**.
Sandbox and sandbox queue reports

Sandbox activity

**Procedure**

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

**Results**

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

Performance Statistics reports

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

**Procedure**

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

The following statistics are available on the Performance Statistics report:

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

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

- MemoryUsed
- GarbageMemory

TurboIntegrator Processes reports

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

**Procedure**

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

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

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

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

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

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

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

   **Completion Status :**
   
   **NORMAL** - Process executed normally
   
   **UNKNOWN** - Process status could not be captured
REJECTED_RECORD - User asked to skip this record and note an error into the log with the ItemReject() TI function

MINOR_ERRORS - Execution finished but had less than the maximum number of minor errors

PROCESS_BREAK - User executed a ProcessBreak() function

ABORTED_BY_PROCESSQUIT - Execution was aborted by a Process Quit() TI function

ABORTED_WITH_ERROR - Process was aborted because a serious error occurred

ABORTED_ON_INIT - Process was aborted because of a serious error during the process startup phase.

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

**Monitoring Chores**

You can monitor the Chores for a serving in the Cognos TM1 Operations Console

**Procedure**

1. If you have not started Performance Monitoring, right-click the server, verify admin access, then right-click the server and click **Start Performance Monitor**.
2. To see the statistics for the Chores running on a server, right-click the server and click **Monitor > Chores**
   The following information is shown for chores: Chore Name, Completion Status, Current State, Client Name, Last Start Time, Last Duration in seconds, Next Activation Time, Current Process.

**Memory Usage Graph**

Displays memory usage as a graphical chart.

Right-click the server you want to use and select **Monitors > Memory Usage Graph**.
Thread Details Graph

The Thread Details Graph displays thread information as a graphical chart.

Right-click the server you want to use and select Monitors > Thread Details Graph.
Monitoring the Cognos TM1 Application Server
You can monitor the server activity of a Cognos TM1 Application Server. Double-clicking on a Cognos TM1 Application Server in the Health tab also provides a report of Tomcat statistics.

Procedure
1. To add a Cognos TM1 Application server to the Cognos TM1 Operations Console, right-click the operation group where you want to list the Cognos TM1 Application Server.
2. Select Add Application Server.
3. Enter a name to use to identify this Application Server and click Create.
4. Right-click the name that you just added and select Configure.
5. Complete the fields on that dialog box:
   - **IP**
     Enter the IP address for the Applications Server. The IP field can also be the fully-qualified domain name or the NetBIOS name.
   - **Context**
     Enter the name of the planning service, for example, pmpsvc
   - **Port number**
     Enter the port number specified in the JVM for TM1. This port number is the jmx port number that you have specified in the applications server JRE options.
   - **Top Refresh period (sec)**
     A typical refresh period is 2.
   - **Top Tolerance Factor (sec)**
     Enter 0 unless you want to expand the tolerance.
   - **Top Time Out (sec)**
     A typical time out value is 1.
6. Click **OK**. If the Cognos TM1 Applications Server is running, the health status becomes green and the activity is being monitored. If the Cognos TM1 Application Server status is not green, ensure that the service is running using IBM Cognos Configuration.

7. Most users prefer to use SSL to securely monitor the TM1 Applications Server. However, you can also monitor it without using SSL. This is a less secure method but does not require the additional certificate steps described in “Using SSL when monitoring the TM1 Applications Server” on page 97. To monitor the TM1 Application Server without using SSL, modify the jvm parameter

```-Dcom.sun.management.jmxremote.ssl=true```

and change it to

```-Dcom.sun.management.jmxremote.ssl=false```

8. Restart the TM1 Applications Server.

**Using SSL when monitoring the TM1 Applications Server**

When using SSL take these additional steps to monitor the Applications Server.

In order to use SSL, you need to create a security certificate on the machine where the Cognos TM1 Application Server is running, then export that certificate to the Cognos TM1 Operations Console machine and configure the Cognos TM1 Operations Console machine so that it uses that new certificate store.

See "Configuring Cognos TM1 Applications to use SSL" in the "Security configurations" chapter of *Planning Analytics Installation and Configuration* for more information.

Depending on whether you are on a 32-bit or 64-bit system, the install location is either:

- `c:\Program Files\ibm\cognos\tm1_64`
- `c:\Program Files\ibm\cognos\tm1`

The location where the jre store is located is either:

- `c:\Program Files\ibm\cognos\tm1_64\bin64\jre\7.0\bin`
- `c:\Program Files\ibm\cognos\tm1\bin\jre\7.0\bin`

The Java certificate store location is either:

- `c:\Program Files\ibm\cognos\tm1_64\bin64\jre\7.0\lib\security\cacerts`
- `c:\Program Files\ibm\cognos\tm1\bin\jre\7.0\lib\security\cacerts`

The certificate store has a default password of changeit. If you have secured your certificate store with another password, use that instead.

On a 64-bit machine there are two JREs shipped with IBM Cognos Cognos TM1

- `install_location\bin\jre\7.0\bin`
- `install_location\bin64\jre\7.0\bin`

This is why there are two cacert stores on 64-bit machines. The 64-bit installation by default runs Apache Tomcat using the jre in the bin64 directory (`install_location\bin64\jre\7.0\bin`) and the instructions therefore add the certificate to the certificate store in the bin64 directory.
You can create the certificate in a different location as long as the Cognos TM1 Application is configured to use that certificate store as described here.

### Creating the SSL certificates

1. Stop the TM1 Application Server if it is running.
2. Use the following command to create a self-signed certificate into your Java store (no line breaks and replace `tm1_64` with `tm1` if you are on a 32-bit machine):

   ```
   keytool -keystore "c:\Program Files\ibm\cognos\tm1_64\bin64\jre\7.0\lib\security\cacerts"
   -alias jmx -genkey -keyalg RSA -dname "CN=${pki-cn}, OU=${pki-ou}, O=${pki-o}, L=${pki-l}, S=${pki-s}, C=${pki-c}" -storepass changeit -keypass changeit
   ```

3. To configure your Cognos TM1 Application Service to use this certificate store add or modify the following Java Version Management (JVM) parameters as appropriate for your installation:

   ```
   -Djavax.net.ssl.keyStore=jre\7.0\lib\security\cacerts
   ```

   **Note:** The port number set here is used later in the process when you configure the Cognos TM1 Applications Server in the TM1 Operations Console: `-Dcom.sun.management.jmxremote.port=7999`

   To update or append these parameters to the JMX_OPTIONS variable in the file, go to the batch file under the bin or bin64 directory called `service_pmpsvc.bat`.

4. Restart the TM1 Application Server from the IBM Cognos Configuration to pick up these changes.

5. Export the certificate from this server store to the machine running Cognos TM1 Operations Console (client) Replace `tm1_64` with `tm1` if needed:

   ```
   keytool -export -alias "jmx" -file jmx -keystore "c:\Program Files\ibm\cognos\tm1_64\bin64\jre\7.0\lib\security\cacerts"
   ```

6. Enter the keystore password: `changeit` to create a cert file called `jmx` in the current directory.

7. Copy the certificate from the server jvm to the client jvm.

8. Use the following command to import ssl to the client (machine running the Cognos TM1 Operations Console) cacert store (replace `tm1_64` with `tm1` if needed):

   ```
   keytool -import -file "jmx" -alias jmx -keystore "c:\Program Files\ibm\cognos\tm1_64\bin64\jre\7.0\lib\security\cacerts"
   ```

9. To configure your Cognos TM1 Application Service to use this certificate store, add or modify the following jvm parameter as appropriate for your installation:

   ```
   -Djavax.net.ssl.keyStore=c:\Program Files\ibm\cognos\tm1_64\bin64\jre\7.0\lib\security\cacerts
   ```

10. If you want to use full path (replace `tm1_64` as needed)

    ```
    -Djavax.net.ssl.keyStore=c:\Program Files\ibm\cognos\tm1_64\bin64\jre\7.0\lib\security\cacerts
    ```

11. Restart the TM1 Application Service from Cognos Configuration.

Follow the previous instructions to configure the Cognos TM1 Application Server in the Cognos TM1 Operations Console.

### Additional configuration information

If Cognos TM1 Application Server and Cognos TM1 Operations Console are on the same Tomcat, you do not need to export and import into the certificate store. The certificate just needs to be created. By default, the pathname of the keystore file where you have stored the server certificate to be loaded is the file ".keystore." It is located in the operating system home directory of the user that is running Tomcat. This is the default store if you run Cognos TM1 Operations Console from a "vanilla" Tomcat. Use the JVM parameter `-Djavax.net.ssl.keyStore=path` to set the certificate with the jmx certificate imported. Set this path for each Cognos TM1 Application Server you want to monitor. Use a different alias for the certificate each time, for example `jmx-frink`. 
Log files for the TM1 Applications server with the TM1 Operations Console

The IBM Cognos TM1 Operations Console can track and monitor activity in the TM1 Applications server.

When you select **Log To Disk** for automatic logging, the rate at which log messages are written is the same as the refresh rate of data in the monitoring window. With **Log to Disk** for Cognos TM1 Applications Server monitoring, the older monitoring data from the log file is overwritten with new log contents.

Unlike configuration of a Cognos TM1 server, you cannot use **Log Append** with the Cognos TM1 Application server. Cognos TM1 Application Server logging can also be scheduled. Two types of Application Server data can be logged:

- Application Server Statistics
- Application Server Session Information

The **Log to Disk** option is available on all monitor windows. Click **Log to Disk** to create an automatic log. The data will continue to be logged until you turn off the logging by unchecking **Log to Disk**, closing the monitor window, or closing the Cognos TM1 Operations Console.

Log files in the IBM Cognos TM1 Operations Console

You can schedule a log, use the log to disk option, or enable automatic logging.

Before any data can display in a log, you must “Verify admin access” on page 91.

You can access these TM1 server logs using the TM1 Operations Console:

- **Transaction log file**
  Displays the transactions recorded in Tm1s.log when a TM1 client changes a cube value.

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

- **Audit log**
  Displays changes to metadata, such as modifications to dimensions, views and subsets.

Right-click the server and select **View Log**, **Message Log**, **Transaction Log**, or **Audit Log** to open these log files. See the "System and Performance Monitoring" chapter of the *TM1 Operations* documentation for details on these logs and how to enable them.

Working with logs

These topics describe how you can adjust the display of log files.

**Configuring the log file and server numeric parameters**

Use the **Configure** option to set the numeric parameters that define how the log file numerics and other server parameters are gathered.

Complete the following parameters for each log file.

- **Log period**
  Specifies the time interval between updates being written to the log file.
  For example, if the screen Refresh is set to 2 seconds, LogPeriod could be set to 10 seconds so that every fifth screen display will be output to decrease the amount of data written to the file.
  Default is 2 seconds.

- **Log Append**
  By default a new log file is not appended, it overwrites the existing log. Select True to append new logfiles to the existing log.

- **Refresh period**
  By default, the log waits 2 seconds before refreshing the data. You can enter a longer or shorter time for the refresh here in seconds.
Tolerance factor
By default, the data is new (0). To permit data in the log to be older, enter a number here in seconds for how old the data can be before requiring a new polling.

Time Out
By default, attempts to connect to the server end after 2 seconds. You can change this number to shorten or lengthen the time the log will keep trying before giving up.

If the IBM Cognos TM1 Operations Console server does not get the updated status in the seconds specified here, then the old data (if it exists) is sent with an indication that a timeout has occurred.

Memory Stats refresh period
By default, the log waits 2 minutes before refreshing the data. You can enter a longer or shorter time for the refresh here in minutes.

Memory Stats tolerance factor
By default, the data is new (0). To permit data in the memory statistics to be older, enter a number here in seconds for how old the data can be before requiring a new polling.

Memory Stats timeout
By default, attempts to connect to the server end after 2 minutes. You can change this number to shorten or lengthen the time the log will keep trying before giving up.

If the IBM Cognos TM1 Operations Console server does not get the updated status in the minutes specified here, then the old data (if it exists) is sent with an indication that a timeout has occurred.

Scheduling logs
You can schedule a log, use the log to disk option, or enable automatic logging.

Procedure
1. To schedule a log, click the Configuration > Logging tab.
2. Depending on the kind of server you want to monitor, click either the TM1 Servers tab or the TM1 Applications tab.
3. Check the server you want to create the log for.
   You may need to drag the pane to reveal the schedule log icons.

4. To create a new log, click Schedule New Log .

Figure 4: Revealing the bottom part of the TM1 Operations Console pane.
5. Define the parameters of the log:

   **Log type**
   Choose one or more kinds of logs: Log Status, Log Enhanced Status (includes object contention columns), Log Sandbox, Log Sandbox Queue

   **Log Duration**
   Enter the Start Date, Start Time, Stop Date, Stop Time, and Log Frequency in seconds. Click in the fields to open a calendar to set the dates.

   **States**
   Check the states to log: Idle, Run, Commit, Rollback, Wait, Login, or Finish.

   **Threads**
   Check the type of threads to log: System Threads, Chores, or User Threads.

6. When the details are defined, click Create.

7. You can use the Filter button to reduce the number of logs shown on the screen.

8. Use 📐 to change the parameters of the log file after it is saved.

### Using Log to Disk to save log files

Log files can be stored using the Log to Disk checkbox that displays on most log screens.

#### Procedure

1. Click the Log to disk option found on most logs.
   Log to disk generates a log of the current screen activity.

2. To see the log, you can right-click the server and select View Log. Check the log you want to see and click OK.

3. Or go to the log location for that server, for example C:\Program Files\IBM\cognos\tm1_64\bin64\opsconsoledata\localhost\sdata\admin\Logs to find the .csv file of the log.

#### Viewing logs

After you have generated a log, use the View Log option to display it.

A user monitoring a server can see all the schedules created by other users. The user can also perform all supported actions on those schedules. However, if the owner deletes the monitored server, that schedule is deleted and becomes unusable by others.

#### Procedure

1. Right-click the server whose logs you want to view.

2. Select View log.

#### Results

The View logs dialog box displays with the list of all logs. The Log Type indicates if the log is scheduled or automatic. You can use the Filter button to restrict the display to only results you are interested in.

### Filtering results in the IBM Cognos TM1 Operations Console

You can reduce the number of entries found in a log or list using filtering.

#### Filtering logs

On the Monitor tab, you can filter the results. When no filter is applied to the results, the filter OFF button is displayed. Click this button to display the list of states you can use to filter the results: Idle, Run, Commit, Rollback, Wait, Login, Finish. Next, select the threads you want to monitor: System Threads, Chores, User Threads.

#### Filtering lists

Use the Filter field and Apply Filter buttons to filter a list of servers or logs. You can type ahead in this field to identify any aspect of the displayed servers or logs.
For example, on the Logging tab you can filter the list of servers.

![Image](image.png)

*Figure 5: Applying filters*

**Exporting logs**
You can export a log file.

**Procedure**
1. Click the **Export** option found on most logs.
2. Select the log content from the window that displays.
3. Paste the content into another file, such as a text file.

**Download a csv version of the log file**
To download a csv version of the log, use the **Download Log File** option.

**Procedure**
1. Right-click the server you want and select **Download Log File**.
2. Click the selection button for the log you want to download.
3. Confirm that you want to download the log and complete the Save or Open dialog box as needed.
4. Browse to the location to store this file and click **Save**.

**Downloading and Uploading the Cognos TM1 Operations Console configuration file**
You can save the details of the Cognos TM1 Operations Console setup using the Upload and Download Configuration File options.

**Procedure**
1. To save the current configuration for the Cognos TM1 Operations Console, click **Download Configuration File** and save the file to the location of your choice.

2. To install the current **tm1opsconsoleconfig.xml** configuration, click **Upload Configuration File**, browse to the location where of the configuration file and click **OK**.
TM1 server Transaction log

The transaction log lists transactions occurring in the specified server over the specified time frame. See the "System and Performance Monitoring" chapter of the TM1 Operations documentation for details on this log and how to use it.

Procedure
1. To get a log of transaction activity, right-click the server and select Transaction log.
2. Click the Start date; Start time; End Date; End time to define the time period over which transactions will be logged.
   If you set just Start time and End time, the time period defaults to all day.
3. You can also use the User, Cube, or Flag filter to identify specific values in each of these columns that you want to use as a filter.
4. To generate a txt file of the log, click Export and select the resulting window's contents. Then you can paste that information into a blank text file.
5. To restore any changed data, click Back Out.
6. When the parameters are set, click OK to retrieve data from the server to the TM1 Operations Console.
   Logging continues until you log out of the session or the end time period is met.

TM1 server Message log

Displays the TM1 server records status messages on the activity of the server in a log file. These messages contain details on activity such as executed processes, chores, loaded cubes and dimensions, and synchronized replication. See the "System and Performance Monitoring" chapter of the TM1 Operations documentation for details on this log and how to use it.

Procedure
1. To get a log of messages, right-click the server and select Message log.
2. Click the Start date; Start time; End Date; End time to define the time period over which transactions will be logged.
   If you set just Start time and End time, the time period defaults to all day.
3. You can also use the Thread, Level, or Logger filter to identify specific values in each of these columns that you want to use as a filter.
4. To generate a txt file of the log, click Export and select the resulting window's contents. Then you can paste that information into a blank text file.
5. You can also use View Process log to see the process log.
6. When the parameters are set, click OK to initiate logging.
   Logging continues until you log out of the session ends or the end time period is met.

TM1 server Audit log

Displays the TM1 server audit log on the activity of the server in a log file. See the "System and Performance Monitoring" chapter of the TM1 Operations documentation for details on this log and how to use it.

Procedure
1. To get a log of the audit activity, right-click the server and select View Audit log.
2. Click the Start Date; Start Time; End Date; End time to define the time period over which transactions will be logged.
   If you set just Start date and End date, the time period defaults to all day.
3. You can also use the Object Type; Object Name; Owner Type; Owner Name; Event Type; Event Code to identify specific values in each of these columns that you want to use as a filter.
4. To generate a txt file of the log, click Export and select the resulting window's contents. Then you can paste that information into a blank text file.
5. You can also use View Detail to see individual log entries.
6. When the parameters are set, click OK to initiate logging.
   Logging continues until you log out of the session ends or the end time period is met.
Using Watchdog to monitor server activity

You can use Watchdog rules to establish a set of criteria that identifies specific states in the servers or processes being monitored and the action to take when the criteria are met. For example, you could create a Watchdog rule that writes to the log when the number of threads reaches 50 on an IBM Cognos TM1 server.

About this task

For a TM1 server, only one Watchdog can be configured by any user having server administrator credentials. It is then available for use by other users. If a user who sets up a server deletes the monitored server, the Watchdog running on that server is deleted and the Watchdog stops running.

Procedure

1. Click 📷.
2. Click the Watchdog tab.
   The tab shows the servers being monitored by the user and the states of the Watchdog rules that have been set up for the servers.
3. Click the server you want to add the Watchdog rule to.
4. To edit the details of the Watchdog rules, you must first verify your admin status. Click Verify and enter the username and password for the administrator user on this server.
   Remember: You may need to drag the pane to expose the rules setup area in the bottom pane.
5. To create a new rule, click 📷.
6. Use the grid to specify when you want a Watchdog event to take place. Click the Field to use for the criteria, such as state; the Operator to apply, such as Equals; and a value.
   For example, you can set State equals Busy. You can use any appropriate value for the fields. The State field accepts these values: Run, Idle, Busy, Commit, Rollback, Logon, Wait, Finish.
7. By default the Action to take when a criteria is met is set to Kill. You can change the action to Log to write the message to the file. Logback can be configured to generate email notifications for these events. See “Sending email alerts with Logback” on page 69.
8. After defining the criteria for the rule, click Save.
9. Add other rules, if needed. Use these icons to work with your rules:
   - Click Edit Rule 📝 to change the criteria of an existing rule.
   - Click Delete Rule ✗ to remove the rule.
   - Click Refresh 🔄 to refresh the list of rules.
10. Set the Frequency for the Watchdog process to run.
11. Click Save to save the Watchdog for this server.
12. To put the Watchdog rule into effect, click Start.
   The Watchdog runs based on the Frequency you entered. If the server meets the criteria of the rule, the action is taken.
13. To stop the Watchdog from running, click Stop.

Example

This example logs an event if the thread count on the TM1 server goes above 20:

<table>
<thead>
<tr>
<th>Threadcount Greater Than 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Equals Log</td>
</tr>
</tbody>
</table>
This example stops any process started by the Admin user after 200 seconds:

- **User** Equals Admin
- **Threshold** Equals 200
- **Action** Equals Kill

This example logs an event if any user other than Admin has been running the TurboIntegrator script called Rogue for over 100 seconds:

- **User** Not Equals Admin
- **Threshold** Equals 100
- **Object** Equals Rogue
- **Action** Equals Log

This example stops a process called Process_MonthlyDataLoad if it was started by a user other than Admin.

- **User** Not Equals Admin
- **Object** Equals Process_MonthlyDataLoad
- **Function** Equals ProcessExecuteEx
- **Action** Equals Kill
Chapter 9. Administering IBM Cognos TM1 Web

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

IBM Cognos TM1 Web Overview

IBM CognosTM1 Web extends the analytical power of TM1 by allowing you to complete the following tasks in a web browser:

- Analyze cube data
- Manipulate formatted Excel reports
- Drill, pivot, select, and filter data
- Build charts
- Perform some server administration tasks

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.

Configuring a proxy account for relational data connections

Configure a proxy account to authenticate TM1 Web connections to relational databases.

Administrators can ensure that, when a TM1 Web user runs an SQL query against a relational datasource, they are not prompted and they are authenticated against a proxy account. The proxy account stores connection information that is stored in the Websheet.

You must use the Java instance that is included in your TM1 installation. Because proxy account configuration uses JDBC, all operating systems are supported.

Note: If no connection information is defined in the Websheet or the user does not enter the correct proxy account information, they are prompted for alternative authentication. For more information, see Viewing relational data in TM1 Web.

You can maintain proxy account information in the relational_host.xml file.

relational_hosts.xml

The relational_hosts.xml file is used to authenticate a TM1 connection to a relational database. It contains the following parameters:

keypath

The location of the relational.key file.

Note: If no path is specified, the relational.key file appears in this location:
**datasource name**
The name given to the data source.

**excelhost**
The relational database name or IP address of the connection defined in the Websheet.

**host**
The relational database name or IP address of the connection to be used on the server.

**Example**

*Note:* This parameter can assist you when you migrate Websheets from development to production. If you define a query in Excel against a database DBDev, the excelhost value should be defined as DBDev. In the development environment, you would set the host to DBDev as well, so that queries in TM1 Web are run against the development relational database server. When you change to the production environment, you would set the host to DBProd, so that queries in TM1 Web are run against the production relational database server. This way, the Websheet database connection does not need to be changed.

**username**
The encrypted user name that is stored in the relational.key file.

**password**
The encrypted password that is stored in the relational.key file.

**RelationalEncryptor.jar**
The RelationalEncryptor.jar file is a command line tool that generates the relational.key file. It also creates or updates entries in the relational_hosts.xml file.

Use the command with the following syntax:

```
java -jar RelationalEncryptor.jar name excelhost realhost username password
```

**relational.key**
The relational.key file contains the encrypted user name and password that appears in the relational_hosts file.

The default location is `tm1web\WEB-INF\cert\key\relational.key`. However, the location can be changed using the keypath parameter in the relational_hosts.xml file.

---

### 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\n```

The following parameters are available.
ActionButtonFullRecalculationEnabled
determines the level of recalculation that occurs as part of the execution of an action button. This parameter is only applicable to action buttons that have Automatically Recalculate Sheet selected as the Calculation type.

If set to true, a full recalculation occurs on the target workbook.

If set to false, a partial recalculation occurs on the target workbook. Only the visible portions of the target workbook are recalculated. This recalculation includes any Active Forms, DBS/DBSW/DBR/DBRW/DBRA/DBSA formulas, and dependencies of cells in the visible area. Any portions beyond the scrolling boundary of the target workbook are not recalculated. False is the default value, which can result in improved performance, especially in large workbooks.

AdminHostName
If set, users are not 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 114.

AdminHostPort
If set, the client tries to use this port instead of the default Admin Host port.

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

CamLoginApiRedirectEnabled
Default value is false.

When enabled, CAM authentication from the TM1 Web API (either URL API or JavaScript Library) performs a redirect to the CAM login page of Cognos Analytics. This behavior differs from the default behavior of showing CAM login page of Cognos Analytics in a dialog box. This parameter must be enabled in cases where Cognos Analytics includes an X-Frame-Options header with a value of SAMEORIGIN or DENY, which is used to improve protection against Click-jacking attacks.

CleanDimensionMetaDataCache
during websheet calculation, the CleanDimensionMetaDataCache parameter specifies whether dimension elements are retrieved from the TM1 server or by using cached elements from TM1 Web.

Default value: false

• If CleanDimensionMetaDataCache is set to false, elements from the tm1web cache are used.

• If CleanDimensionMetaDataCache is set to true: tm1web dimension elements are cleaned from the cache and the elements are retrieved directly from the TM1 server.

CrossDomainAccessList
Specifies a list of cross-domain URLs that are allowed to access TM1Web.

You can use this parameter to specify the domain where IBM Cognos Workspace is running, if it’s running on a domain separate from TM1 Web.

Use an asterisk (*) to allow any domain to access TM1 Web.

If you specify multiple URLs, separate each one by using a comma.

If this parameter is not set or the parameter value is empty, no cross-domain access to TM1 Web is allowed.

CubeViewerColumnPageSize
Specifies the number of columns to fetch in a page of Cubeviewer.

See “Changing the Cube Viewer Page Size” on page 122.

CubeViewerHiddenDimensionsEnabled
Allows you to hide dimensions in the TM1 Web cube viewer.

Hidden dimensions are part of the context of a view, but do not show up as context dimensions in the TM1 Web cube viewer. Instead, they reside in a region of the dimension bar labeled Hidden.
To use hidden dimensions in the TM1 Web cube viewer, you must set CubeViewerHiddenDimensionsEnabled="true" in the tm1web_config.xml file. When the feature is enabled, the Hidden region appears on the cube viewer.

You can drag and drop dimensions to and from the Hidden region just as you can for the Rows, Columns, and Context regions.

When a view includes hidden dimensions, the number of hidden dimensions is displayed below the Hidden label. When you click the Hidden region, you can see which dimensions and elements are hidden.

You cannot change the element for a hidden dimension. If you want to change an element, you must show the dimensions by dragging it to the Rows, Columns, or Context region, and then change the element. You can then return the dimension to the hidden region.

CubeViewerPageSize
   Specifies the number of rows to fetch in a page of Cubeviewer.

   See “Changing the Cube Viewer Page Size” on page 122.

CubeviewerStringWrap
   Settings for string cell wrapping in the Cubeviewer.

   See “Wrapping string values in cube views” on page 122.

CustomCAMLogoutUrl
   Specifies the URL of a dedicated Logout page for CA SiteMinder when TM1 is configured to use CAM security (mode 4 or 5). This Logout page must be accessed on logout so that the SiteMinder session cookie can be invalidated.

   When a user clicks Logoff in TM1 Web, the CAM logout occurs first. Then, the SiteMinder Logout page is called.

EvaluationServiceURL
   Specifies the location of the evaluation service. Valid value is hostname:port_number. If no value is assigned, the location is assumed to be http://localhost:9510.

ExportCellsThreshold
   Specifies the maximum number of cells that an export of a Websheet or a cube view can contain. If the number of selected cells exceeds the threshold, a warning message is displayed and the export does not start.

   Edit the ExportCellsThreshold parameter in the tm1web_config.xml file by using the following format:

   

   `<add key="ExportCellsThreshold" value="CellsThreshold" />`

   where CellsThreshold is the cell count threshold determined by multiplying the number of rows by the number of columns per sheet, and then multiplying that result by the number of iterations and context members that the export is selected for.

   For example, if a Websheet has two sheets and each sheet has 1000 rows and 25 columns, and the export is selected for four context members, the cell count is calculated as 25,000 * 2 sheets * 4 context members = 200,000 cells. If the <CellsThreshold> is 150,000, this Websheet export would be rejected.

ExternalUrl
   Set the ExternalUrl parameter if you are using TM1® Web and Cognos security (CAM) authentication with an external load balancer that modifies the original startup URL for TM1 Web. The ExternalUrl parameter provides the correct URL so that Cognos security can successfully redirect back to TM1 Web.

   Set the value to the same URL that you use to start TM1 Web, for example

   `<add key="ExternalUrl" value="http://mycomputer/TM1Web" />`

GzipCompressionEnabled
   Determines if the web server responses will be compressed. Valid values are true/false.

HideCubeviewerToolBar
   If set to true, all Cubeviewer toolbar are not displayed.

   See “HideCubeviewerToolBar Parameter” on page 121.
**HideTabBar**
If set to true, multiple tabs are not displayed.
See “HideTabBar Parameter” on page 120.

**HideWebsheetToolBar**
If set to true, all websheet toolbars are not displayed.
See “HideWebsheetToolBar Parameter” on page 120.

**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 117.

**LegacyUrlApiSessionDiscoveryEnabled**
Use the LegacyUrlApiSessionDiscoveryEnabled configuration parameter to control how the TM1 Web URL API handles login sessions. Configure this parameter to specify whether or not the URL API tracks separate unique login sessions.

This parameter enables the URL API session to be reused based on the specified admin host, TM1 server, and (optional) user name.

If you are using the session token login approach with the URL API, you must set the LegacyUrlApiSessionDiscoveryEnabled configuration parameter in the tm1web_config.xml file to False. For more information about logging in with a session token, see TM1 Web API session login.

Use this format:

```xml
<add key="LegacyUrlApiSessionDiscoveryEnabled" value="True or False"/>
```

For example:

```xml
<add key="LegacyUrlApiSessionDiscoveryEnabled" value="False"/>
```

The default value is True.

- **True**
  TM1 Web tries to match new login request with an existing login session based on the provided information (TM1 Admin host, TM1 Server, user name).
  This parameter should only be set to True if a single login will occur for a unique TM1 Admin Host, TM1 server, and user name combination.

- **False**
  Specifies that a session token must be provided every time that you open a TM1 Web object with the TM1 Web URL API. Otherwise, the user is prompted.
  Set this parameter to False if you plan to use multiple login sessions with TM1 Web URL API. You also use this configuration if you are using multiple login sessions with the URL API and other TM1 Web clients such as TM1 Web and TM1 Application Web. This configuration uses the session token to keep the user sessions separate and unique.

**MaximumConcurrentExports**
Determines the maximum number of concurrent exports that can be executed from TM1 Web. The default value is 5.

You can set MaximumConcurrentExports to 0 to allow an unlimited number of concurrent exports. This setting is analogous to export behavior in TM1 Web before version 10.3.

If the maximum number of concurrent exports is reached, and additional exports are then initiated, the additional exports are queued until an export slot is available. The initiator of a queued export does not receive notification of queuing.

The optimal parameter setting depends on your RAM capacity and your user requirements. Generally, the more RAM you have available to TM1 Web, the higher the parameter setting can be. Increasing the value results in increased memory consumption, but reduces export queuing. (Setting the parameter to 0 eliminates export
queuing.) Conversely, decreasing the parameter value reduces memory consumption that results from exports, but can result in more frequent export queuing.

**MaximumSheetsForExport**
Specifies the maximum number of sheets that are allowed to export.
See “Setting the Maximum Number of Sheets to Export from a Cube Viewer” on page 122.

**MixedCellPaste**
If the MixedCellPaste parameter is set to true, when you copy values to a mixed range of leaves and consolidated values in a Websheet, the pasted values will match exactly.

**Note:** This parameter applies to Websheets only; it does not apply to CubeViewer.

The default value is false.

**NavTreeCollapsedOnStart**
Determines whether the navigation panel will be collapsed or expanded after a user logs in.
See “NavTreeCollapsedOnStart Parameter” on page 120.

**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 121.

**NavTreeHidden**
Determines whether the navigation panel will be displayed after a user logs in.
See “NavTreeHidden Parameter” on page 119.

**RecalcOnActivate**
If RecalcOnActivate is set to true, a recalculate is performed each time a websheet or cubeview is activated in TM1 Web, for example, when you switch tabs.

Valid values are true or false.

**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.

**RelationalResultMaxRows**
If a value greater than -1 is specified, then relational query ResultSets are limited to returning the specified number of rows.

**TM1DatabaseLabel**
If set to "Y", the name of the database is displayed beside the user on the TM1 Web banner. For example, "Welcome: Admin / Planning Sample". The default is "N". When this option is set to "N", nothing is displayed beside the user.

See “TM1DatabaseLabel Parameter” on page 121 in Configuring IBM Cognos TM1 Web Startup and Appearance Settings.

**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 114.
**UseBookRecalcSetting**

The UseBookRecalcSetting parameter is included in the `tm1web_config.xml` file. When set to true, the web server honors the mode in which the Excel sheet was published. If the Excel sheet was published in Manual recalc mode, websheet data is not resent to the client until a recalculation is performed.

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

```xml
<add key="UseBookRecalcSetting" value="false" />
```

where value is either "false" or "true"

If you set UseBookRecalcSetting to true, TM1 Web honors the recalculation settings in the Excel worksheet.

When Calculation Options is set to Automatic:

- If you set UseBookRecalcSetting = "true", the websheet is recalculated automatically when you change the SUBNM function.
- If you set UseBookRecalcSetting = "false", the websheet is recalculated automatically when you change the SUBNM function.

When Calculation Options is set to Manual:

- If you set UseBookRecalcSetting = "true", the websheet is not recalculated automatically. To recalculate, you must manually click the recalc button.
- If you set UseBookRecalcSetting = "false", the websheet is recalculated automatically when you change the SUBNM function.

**WebsheetBackgroundRecalculationMode**

Specifies the level of background recalculation that occurs for a websheet.

WebSheetService.scrollWebSheet calls can take several seconds because the data is not readily available. Use the WebsheetBackgroundRecalculationMode parameter to recalculate the book in the background so that the necessary data is ready when it is requested.

If set to 0 (default value), only the buffered (visible) area is calculated on a refresh of a sheet.

If set to 1, the area that is adjacent to the buffered area is calculated, in addition to the buffered area. This improves wait times if the user scrolls slightly away from the initially visible area.

If set to 2, the entire current worksheet is calculated. This improves wait times if the user scrolls to any area of the current sheet.

If set to 3, the entire current workbook is calculated. This improves wait times if the user moves to any area of the current worksheet or to another worksheet.

**Note:** The higher the setting number, the more cells are calculated meaning that there would be a higher load on the web server.

**WorkbookMaxCellCount**

Specifies the maximum cell count of a workbook as a number with no thousands separators.

The TM1Web application server validates the size of a workbook that is published to TM1 server. Workbooks that contain ActiveForms might be uploaded only with their master row. At publish time, the workbook can have multiple rows but when it is opened and rebuilt it can display many more rows. You can use WorkbookMaxCellCount to avoid issues opening workbooks with many cells.

If this parameter is present in `tm1web_config.xml` and it is not the default, when the user opens a workbook, the server validates its cell count against WorkbookMaxCellCount. If the cell count of the workbook exceeds WorkbookMaxCellCount, an error message is logged and the workbook is not opened. The user sees the `<book_name> exceeds maximum cell count` error message in the `tm1web.log` file. For more information, see Using IBM Cognos TM1 Web Logging.

- Leaving this parameter blank or setting it to less than 0 indicates that an unlimited cell count for workbooks is allowed.
- The default value is -1, which indicates an unlimited number of cells are allowed in a workbook.
• Setting this parameter to 0 indicates that workbooks cannot have any cells. Therefore, anything above 0 is recommended.

**Note:** Changes to this parameter require a restart of the application server.

**X-Frame-Options**

The X-Frame-Options parameter sets the X-Frame-Options response header value. The parameter (and the response header value) specifies whether a browser should be allowed to render a TM1 Web page in a <frame>, <iframe>, or <object>. Use this parameter to prevent Click-jacking attacks and ensure that TM1 Web content is not embedded into other sites. There are three possible parameter values.

- **0** corresponds to the DENY response header value, which prevents any domain from framing TM1 Web content.
- **1** corresponds to the SAMEORIGIN response header value, which allows only the current domain to frame TM1 Web content.
- **2** corresponds to the ALLOW-FROM response header value. In this case, TM1 Web checks the CrossDomainAccessList parameter in tm1web_config.xml for the list of cross-domain URLs that are allowed to access and frame TM1Web content.

The ALLOW-FROM response header does not have universal browser support. TM1 Web uses the values in CrossDomainAccessList to determine whether the domain is allowed or not. If not, TM1 Web includes the DENY response header value, which prevents framing. In certain circumstances, TM1 Web might be unable to determine the requesting domain. In this case, the SAMEORIGIN response header value is included.

If the X-Frame-Options parameter is missing or empty, 2 is the default value.

The .jsp files in TM1Web include the response header X-Frame-Options only for the DENY and SAMEORIGIN values. If the domain is confirmed to be allowed, then no X-Frame-Options header is included.

**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 Word Pad 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\`

   **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 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.
- If the AdminSvrSSLCertID parameter is incorrectly configured, the server name pull-down displays as empty and an error is logged in the Cognos TM1 Web log file. For more information, see Running TM1 in Secure Mode using SSL in TM1 Operation.

After the user enters a valid User Name and Password, IBM Cognos TM1 Web will log in 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 box, 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 box 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 the server and select Security, Clients/Groups.
   The Clients/Groups dialog box opens.

2. Click Settings.
   The Client Settings dialog box 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 box 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 box.

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 119.

   The available settings for controlling the appearance of the Navigation pane include:
   • Include the Navigation Pane - Determines whether 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 whether 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 box. The number of clients selected in the Subset Editor is summarized in the Client Settings dialog box.

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 box.
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 116.

The `HomePageObject` parameter works for three types of objects:

- Cubeviewer
- Websheet
- URL

The homepage object displays after the user successfully logs in to IBM Cognos TM1 Web.

**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
&ltadd 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.

![Figure 6: Example of an assigned name for an uploaded Excel file in Server Explorer](image)

2. Set the value parameter using the following format:

```
value="TM1://ServerName/blob/PUBLIC/\}Externals\TM1_Filename"
```

where:
- `ServerName` is the name of the TM1 sever 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" />
  ```

### 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 115.

#### 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:

```
<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 cannot 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.

Example of `HideTabBar` parameter set to false

![Example of HideTabBar parameter set to false](image)

*Figure 7: Example of HideTabBar parameter*

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:

```xml
<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.

   ```xml
   <!--NavTreeDisplayServerView: Y/N - Whether 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.

**TM1DatabaseLabel Parameter**

This parameter displays the TM1 database label in the banner beside the user name.

Edit the `TM1DatabaseLabel` parameter in the `tm1web_config.xml` file using the following format:

```xml
<add key="TM1DatabaseLabel" value="Y"/>
```

where `TM1DatabaseLabel` can be either N or Y.

- If `TM1DatabaseLabel` is set to N, the database label is not displayed. This is the default behavior of IBM Cognos TM1 Web.
- If `TM1DatabaseLabel` is set to Y, the database label appears in beside the logged in user name in the banner as "Welcome: <user name> / <TM1 database label>".
Changing the Cube Viewer Page Size
You can change the number of rows and columns displayed in the Cube Viewer of IBM Cognos TM1 Web.
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.

```properties
# System logging settings
log4j.rootLogger=ERROR, TextFile
log4j.logger.com.ibm.cognos=ERROR
log4j.logger.com.cognos=ERROR
log4j.logger.com.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 versions 10.2.0 and later use 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 .xlsm 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.

Checking default font settings for non-Microsoft Windows web servers

For non-Microsoft Windows web servers, check what fonts you have available on your web server and choose one of these fonts as the default font for Microsoft Excel. Use the default font for creating any Microsoft Excel workbooks that you use in TM1 Web.

Typically a Microsoft Windows web server has the fonts that are used by Microsoft Excel, but this might not be the case for non-Microsoft Windows web servers.

If you are noticing differences between the column widths that you see in Microsoft Excel and what you see in TM1 Web, it is likely because the font you are using in Microsoft Excel is not available on your web server.

On AIX, TM1 looks for fonts in the following location: /usr/lpp/X11/lib/X11/fonts/TrueType.

Column width measurements are based on the default font that is set in Microsoft Excel. The default font is set under Options, General, Use this as the default font. If you see Body Font, this is typically the default Microsoft Excel font, Calibri.

You can verify what font is being used as the default font in a workbook. Extract the workbook. Under the x1 folder, find the styles.xml file. Open the file in a text editor and look for the following fonts section:

```xml
<fonts count="2" x14ac:knownFonts="1"><font><sz val="11"/><color theme="1"/><name val="Calibri"/><family val="2"/><scheme val="minor"/></font>
```

Unless you have a Calibri TrueType font for AIX, change the default font in Microsoft Excel to Lucida Sans or another font that is available on your AIX web server.
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:

<table>
<thead>
<tr>
<th>Base Cube</th>
<th>Dependent Cube</th>
</tr>
</thead>
<tbody>
<tr>
<td>This cube has values from Cube A or depends on it.</td>
<td>This cube has values that the Base Cube is using.</td>
</tr>
</tbody>
</table>

In this simple case, the caches in Base cube are invalided when:

- there is a change in the Base cube itself
  
or
  - there is a change in the dependent cube

Since a dependency is established, when there is a change in the dependent cube, the Base cube must also change.

Without this dependency map, the Base cube data could become inconsistent with the Dependent cube. The Base data is dependent upon the data in the dependent cube so any change in the Dependent cube requires the Base cache to change also.

Changes in Dependency Mapping

This topic describes cube dependency mapping.

**Note:** If you are upgrading from TM1 9.4.1 FP2 and earlier, there is a change you should note. In previous releases, the 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 two rules: one that pulls from Cube C and the other pulls from Cube D. Cube D has a single rule that pulls from Cubes E and F.

When User 1 reads from Cube B, a rule calculated there requires data pulled from Cube C. So Cube B, C and related cube objects are blocked until User 1’s query is completed.

When User 2 reads from Cube B, a rule calculated there requires data pulled from Cube D. Cube B, D, and related cube objects are now blocked until User 2’s query is completed.

To address this situation, the TurboIntegrator function AddCubeDependency was created to establish dependencies. This function allows users to establish dependencies after events such as a re-start of the server.

### Clearing Cube Dependencies

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

<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 casues 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</td>
<td>Calculated Feeders will create dependencies.</td>
</tr>
<tr>
<td>Event</td>
<td>Dependency Action</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Server Startup + Persistent Feeders</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.

This function can be set in anywhere in the TurboIntegrator Advanced tab.

**Debugging and Identifying Cube Dependencies**

Using TurboIntegrator to preset dependencies may be necessary.

To identify existing cube dependencies requires an in-depth understanding of an application so that you can analyze all of the rules for each cube including control cubes. Even when you have that level of knowledge about an application, it is important to turn on the following logging capability to identify cube dependencies as they are established.

For example, when the following command is added to the tm1s-log.properties file:

```java
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:

```java
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 135.
- **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 41 in the IBM Cognos TM1 Operations documentation.

- **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.

![Diagram of replication connections between source and target servers](image)

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 138.

**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 135.

For details on the Admin Server, see “TM1 Admin Server” on page 9.

**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 136.
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 138.
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 Planning Analytics Installation and Configuration.

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 138.

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 138.

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 cannot 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 137.

5. Click OK.

   TM1 replicates the cube, as described in “What Happens when TM1 Replicates a Cube” on page 138.

   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 41 in the IBM Cognos TM1 Operations documentation.

If there is an update conflict for a particular cell, TM1 uses the most recent update, no matter where this update was made (that is, in the source cube or the mirror cube).

When a synchronization occurs, either as the result of a scheduled chore or manual execution, a log file that records all the actions that occur during the synchronization is created in the logfiles directory on the target TM1 server. The synchronization log file is named TM1ReplicationLog_<time stamp>_<source server>.log. For example, TM1ReplicationLog_20140308035140_SData.log.

The synchronization log records the time at which each action within the synchronization occurs, the name of the user who established the synchronization, and a brief description of the action.

20140308041525 [Admin] => Synchronization to "Planning Sample" started.
20140308041525 [Admin] => Current R&S values: LastSyncTime=20140308041233 LastSyncTimeMaster=20140308041233 LastSyncStarRecord=20140308041233.
20140308041525 [Admin] => Start: Replication of objects not yet copied.
20140308041526 [Admin] => Replicating dimension "plan_version".
20140308041526 [Admin] => Synchronizing cube: (P)"plan_BudgetPlan" => (S)"plan_BudgetPlan"
20140308041526 [Admin] => WARNING: No changes found in Planet server.
20140308041526 [Admin] => Synchronization of cube "plan_BudgetPlan" completed: Star <= 0
20140308041526 [Admin] => Synchronizing cube: (P)"promotions" => (S)"promotions"
20140308041528 [Admin] => Synchronizing of cube "promotions" completed: Star <= 12
20140308041528 [Admin] => Synchronizing cube: (S)"plan_BudgetPlan" => (P)"plan_BudgetPlan"
20140308041528 [Admin] => WARNING: No changes found in Star server.
20140308041528 [Admin] => Synchronization of cube "plan_BudgetPlan" completed: Planet <= 0
20140308041528 [Admin] => Synchronizing cube: (S)"promotions" => (P)"promotions"
20140308041528 [Admin] => Synchronization of cube "promotions" completed: Planet <= 30
20140308041530 [Admin] => Synchronization to "Planning Sample" completed.
LastSyncTime=20140308041527 LastSyncTimeMaster=20140308041526 LastSyncStarRecord=20140308041234.

**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 documentation.

Procedure
1. In the Server Explorer, select the Chores icon beneath the target server for which you want to automate synchronization.
2. Right-click the Chores icon and click Create New Chore.
   The Chore Setup Wizard opens.
3. Select the replication for which you want to automate synchronization.
4. Click the arrow button to move the replication from the Available list to the Selected List.
5. Click Next.
   The second screen of the Chore Setup Wizard displays.
6. Use the calendar and Time field to set a start date and time for the initial synchronization.
7. Use the fields in the Chore Execution Frequency box to set the interval at which the synchronization should be executed.
8. Click Finish to complete the scheduling.

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 Planning Analytics Installation and Configuration.
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>
<thead>
<tr>
<th>Table 6: Keyboard shortcuts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action</strong></td>
</tr>
<tr>
<td>Perform the command for an active command button</td>
</tr>
<tr>
<td>Close an object or pane that is closable</td>
</tr>
<tr>
<td>Go to the first item or object; Go Home</td>
</tr>
<tr>
<td>Go to the last item or object; Go to the End</td>
</tr>
<tr>
<td>Move forward through the panes of the application</td>
</tr>
<tr>
<td>Move backward through the panes of the application</td>
</tr>
<tr>
<td>Move the focus to the Application Bar (blue dot)</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>
</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>
</tr>
<tr>
<td>Toggle on or off</td>
</tr>
<tr>
<td>Move to the next option button and select it</td>
</tr>
<tr>
<td>Move to the previous option button and select it</td>
</tr>
<tr>
<td>Open and display the contents of a drop-down list</td>
</tr>
<tr>
<td>Close an open drop-down list</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>
</tr>
<tr>
<td>Move to the previous selectable node</td>
</tr>
<tr>
<td>Expand the current selection if it is not expanded. If the node is expanded, go to the first child node</td>
</tr>
<tr>
<td>Collapse the current selection if it is expanded. If the node is collapsed, go to the parent node before the current selection</td>
</tr>
<tr>
<td>Expand the children menu items</td>
</tr>
<tr>
<td>Collapse the children menu items</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Open a context menu</td>
</tr>
<tr>
<td>Close an open context menu</td>
</tr>
<tr>
<td>Scroll down</td>
</tr>
<tr>
<td>Scroll up</td>
</tr>
<tr>
<td>Move to the next widget in the tab index order at the same level in the canvas</td>
</tr>
<tr>
<td>Move to the previous widget in the tab index order at the same level in the canvas</td>
</tr>
<tr>
<td>Add a database instance</td>
</tr>
<tr>
<td>Upload a configuration file</td>
</tr>
<tr>
<td>Download a configuration file</td>
</tr>
<tr>
<td>Tile vertically</td>
</tr>
<tr>
<td>Tile horizontally</td>
</tr>
<tr>
<td>Box tile</td>
</tr>
<tr>
<td>Refresh the tree</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 155.

To toggle the display of control cubes and other control objects in Server Explorer, click **View, Display Control Objects**. All control cubes appear in Server Explorer with a prefix of a right curly brace ( { }). For example, }CellSecurity_SalesCube.

Security Control Cubes

Security control cubes apply security privileges for TM1 objects to user groups on the TM1 server.

Most of these control cubes are populated with the privileges assigned in the TM1 Security Assignments window, but you can also apply privileges directly in the control cubes. However, you cannot apply privileges to the ADMIN group; this group always has ADMIN privileges to all objects on the TM1 server.

}{CellSecurity_CubeName

TM1 creates an empty }CellSecurity_CubeName control cube when you initiate the process of defining the cell-level security.

To do so, select a cube in Server Explorer, and click **Security, Create Cell Security Cube**. You can use the control cube to define the cell-level security for the selected cube. TM1 applies the cell-level security to the user groups.

The cells in a }CellSecurity_CubeName control cube can contain one of the following three strings, which correspond to the security privileges that you can assign to the cells:

- Read
- Write
- None

For details on defining the cell-level security and for definitions of the security privileges, see IBM Cognos *TM1 for Developers*.

Dimensions

A }CellSecurity_CubeName cube contains all the dimensions that are present in the cube for which you want to define the cell-level security. In addition, a }CellSecurity_CubeName cube contains the following dimension:

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

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 IBM Cognos TM1 for Developers.

**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 for Developers.

**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 IBM Cognos TM1 for Developers.

**Dimensions**

The }DimensionSecurity control cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>}Dimensions</td>
<td>Dimensions available on the TM1 server.</td>
</tr>
<tr>
<td>}Groups</td>
<td>Groups defined for the TM1 server.</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 IBM Cognos TM1 for Developers.

**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>]Groups</td>
<td>Groups defined for the TM1 server.</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>}Processes</td>
<td>All processes defined on the TM1 server.</td>
</tr>
<tr>
<td>]Groups</td>
<td>Groups defined for the TM1 server.</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 IBM Cognos TM1 for Developers.

The cells in the ]ClientsGroups cube contain strings that assign clients to one or more of the user groups available on the TM1 server.

In this example:
- Admin client is assigned to the ADMIN group
- Usr1 client is assigned to the North America group
- Usr2 client is assigned to the South America group
- Usr3 client is assigned to both the North America and South America groups
The \( \text{ClientsGroups} \) control cube contains the following dimensions:

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

\( \text{ClientProperties} \)

The \( \text{ClientProperties} \) 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 \( \text{ClientProperties} \) 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 \( \text{ClientProperties} \) 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 \( \text{ClientProperties} \) 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.

The \( \text{ClientProperties} \) control cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Clients} )</td>
<td>Clients currently defined for the TM1 server.</td>
</tr>
<tr>
<td>( \text{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 155.

**Object Attribute and Property Control Cubes**

The object attribute and property control cubes store attribute and property values for objects on the TM1 server.

\( \text{ConnectionProperties} \)

The \( \text{ConnectionProperties} \) 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. For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 155.</td>
</tr>
</tbody>
</table>
**Dimensions**

The }DimensionProperties cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td>Dimensions available on the TM1 server</td>
</tr>
<tr>
<td>}DimensionProperties</td>
<td>Properties that can be set for dimensions. For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 155.</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>Dimensions</td>
<td>Dimensions available on the TM1 server</td>
</tr>
<tr>
<td>}DimensionAttributes</td>
<td>Attributes that have been defined for dimensions on the TM1 server.</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>}ElementAttributes_DimensionName</td>
<td>Element attributes defined for the DimensionName dimension. For details on defining element attributes, see IBM Cognos TM1 for Developers.</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 for Developers* documentation.

### 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>
<tr>
<td>}HierarchyProperties</td>
<td>Includes the following two types of elements</td>
</tr>
<tr>
<td></td>
<td>• A single element, named defaultMember, for assigning the default member of the dimension.</td>
</tr>
<tr>
<td></td>
<td>• A set of 21 elements (level000 - level020) for assigning names to the dimension hierarchy levels.</td>
</tr>
<tr>
<td></td>
<td>For more details, see “}HierarchyProperties” on page 160.</td>
</tr>
</tbody>
</table>

### 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 3, “Remote Cognos TM1 Server Operations,” on page 13.

#### }StatsByClient

For each client on the server, the }StatsByClient control cube tracks the message count, average message size, total elapsed time, and other measures.

### Dimensions

The }StatsByClient cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>}StatsStatsByClient</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 Appendix C, “Control Dimensions,” on page 155.</td>
</tr>
</tbody>
</table>
### PerfClients
Each client on the server, plus the consolidated element Clients Total, which is a consolidation of all clients.

### TimeIntervals

### 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>StatsStatsByCube</td>
<td>This dimension contains the following measures:</td>
</tr>
<tr>
<td></td>
<td>Memory Used for Views: Counts views which have been opened. Counts all views, whether private or public, regardless of user.</td>
</tr>
<tr>
<td></td>
<td>It is NOT dependent on who happens to be logged in.</td>
</tr>
<tr>
<td></td>
<td>Number of Stored Views: Counts all views which have been opened, whether private or public, regardless of user. Includes all views which have been looked at.</td>
</tr>
<tr>
<td></td>
<td>Number of Stored Calculated Cells</td>
</tr>
<tr>
<td></td>
<td>Number of Populated String Cells</td>
</tr>
<tr>
<td></td>
<td>Number of Populated Numeric Cells</td>
</tr>
<tr>
<td></td>
<td>Number of Fed Cells</td>
</tr>
<tr>
<td></td>
<td>Memory Used for Calculations</td>
</tr>
<tr>
<td></td>
<td>Memory Used for Feeders</td>
</tr>
<tr>
<td></td>
<td>Memory Used for Input Data</td>
</tr>
<tr>
<td></td>
<td>Total Memory Used</td>
</tr>
<tr>
<td>PerfCubes</td>
<td>This dimension contains numeric elements for each cube on the server, plus the consolidated element Cubes Total, which is a consolidation of all clients.</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>PerfClients</td>
<td>Each client on the server, plus the consolidated element Clients Total, which is a consolidation of all clients.</td>
</tr>
<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, “Control Dimensions,” on page 155.</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, “Control Dimensions,” on page 155.</td>
</tr>
</tbody>
</table>

**StatsByRule**

The }StatsByRule control cube stores performance statistics for cube rules on the TM1 server.

To enable the storage of statistics for a rule, you must set the Rule_Stats property for the associated cube to YES in the }CubeProperties control cube.

Each time a rule is changed or compiled, the data for that rule is cleared and updated in the }StatsByRule control cube. This helps you to immediately see the impact of a rule change.

The data in the }StatsByRule control cube does not persist between server sessions, it is cleared every time that you restart your TM1 server.

The }StatsByRule control cube contains the following dimensions:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubes</td>
<td>Each cube on the TM1 server is included as an element in the }Cubes dimension. To view statistics for a different cube rule, select a new cube name from the }Cubes dimension.</td>
</tr>
<tr>
<td>LineNumber</td>
<td>Numeric elements 1 through 10,000. The elements correspond to line numbers in a TM1 rule.</td>
</tr>
<tr>
<td>Note:</td>
<td>The Rules Editor in TM1 Architect and Perspectives does not indicate line numbers. To view a rule with line numbers, open the cube_name.rux file in a text editor that includes line numbers. The Rules Editor in TM1 Performance Modeler does include line numbers.</td>
</tr>
<tr>
<td>Dimension</td>
<td>Elements</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| }RuleStats    | Contains elements that correspond to the information and statistics that are stored for rules on your TM1 server, including: • Rule Text  
|               | • Total Run Count  
|               | • Min Time  
|               | • Max Time  
|               | • Avg Time  
|               | • Total Time  
|               | • Last Run Time  
|               | For details, see "}RuleStats" on page 161 control dimension.                                                                                                                                              |

The times recorded for Min Time, Max Time, Avg Time, Last Run time, and Total Time are in milliseconds (one one-thousandth of a second). Some rules execute faster than 1 millisecond, resulting in an entry of 0 for the time. It's possible for a simple rule to run multiple times, while the Total Time shows as 0.

<table>
<thead>
<tr>
<th>StatsForServer</th>
<th>The }StatsForServer control cube tracks the connected clients, active threads, and memory used for the server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td>The }StatsForServer cube contains the following dimensions:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
</table>
| StatsStatsForServer | This dimension contains the following measures:  
|                  | Number of Connected Clients  
|                  | Number of Active Threads  
|                  | Memory Used (bytes)  
|                  | Memory in Garbage (bytes)  
|                  | For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 155.                                                                                                               |


**Other Control Cubes**

The following sections describe all other control cubes available on the TM1 server.

<table>
<thead>
<tr>
<th>HOLD_UserName_CubeName</th>
<th>The }Hold_UserName_CubeName control cube tracks the cells held by the UserName client on the CubeName cube.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td>The }Hold_UserName_CubeName cube contains the following dimensions:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Dimension

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Elements</th>
</tr>
</thead>
</table>
| )Hold     | This dimension contains the following elements:  
|           | OriginalVal |
|           | Hold Status |
|           | For complete details on this dimension, see Appendix C, “Control Dimensions,” on page 155. |

### 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 yyyymmdhhss.</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 yyyymmddhhss.</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 yyyymmddhhss.</td>
</tr>
<tr>
<td>LASTSYNCSTARRECORD</td>
<td>Time (GMT) at which the most recent synchronization resulted in data moving from the star server to the planet. The time format is yyyymmddhhss.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>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.</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Stores and displays the password in an encrypted format for establishing a replication connection.</td>
</tr>
<tr>
<td>SYNCPLANET</td>
<td>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.</td>
</tr>
<tr>
<td>SYNCSTAR</td>
<td>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.</td>
</tr>
<tr>
<td>USER</td>
<td>For a given replication connection, stores the name of the user who established the connection.</td>
</tr>
</tbody>
</table>

**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 `ConnectionProperties` control cube.

**CubeFunctions**

The `CubeFunctions` dimension is used only in the `StatsByCubeByClient` control cube.

The dimension contains the following string elements corresponding to functions that can be performed on a cube:

<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>
</tbody>
</table>
### Cell Update

Cell updates include user input through the Cube Viewer, In-Spreadsheet Browser, TM1 Web, and slices. Cells updated through rule calculations or other methods are not included in this metric.

### View Calculation

Number of views requested, which can be user requests for Cube Viewer or slices.

### View Retrieval

Number of views requested and displayed.

---

### CubeProperties

The CubeProperties dimension is used only in the CubeProperties control cube. The dimension contains the following string elements corresponding to the properties that can be defined for cubes on the IBM 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>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>Rule_Stats</td>
<td>This property determines if performance statistics are collected for the rules associated with the cube. To enable statistic collection, set the Rule_Stats property to YES. To disable statistic collection, set the property to NO (this is the default value). Rules statistics are stored in the StatsByRule control cube. Rule_Stats is a dynamic property, meaning that it does not require a server restart to take effect. However, it can take up to 60 seconds for a dynamic property change to be applied on the TM1 server.</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>Element</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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. The value of VMM is expressed in kilobytes. If no VMM value is specified the default value is 128 kilobytes. The valid range for VMM is 0 - 2,147,483,647 KB. The actual upper limit of VMM is determined by the amount of RAM available on your system.</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. For details on stargate views, see “About Stargate Views” on page 25.</td>
</tr>
</tbody>
</table>

**Cubes**

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

**DimensionAttributes**

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 }DimensionAttributes control cube.

**DimensionProperties**

The }DimensionProperties dimension is used only in the }DimensionProperties control cube. The }DimensionProperties dimension contains the following string elements that correspond to the properties that can be defined for dimensions on the IBM 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>Element</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Last_Time_Updated</td>
<td>Time (GMT) at which a dimension was last updated. The time format is yyyymmddhhss.</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>
<tr>
<td>SortElementsType</td>
<td>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.</td>
</tr>
<tr>
<td>SyncAttributes</td>
<td>For replicated cubes, this property indicates whether element attributes are synchronized when the associated dimension is synchronized.</td>
</tr>
<tr>
<td>SyncSubsets</td>
<td>For replicated cubes, this property indicates whether subsets are synchronized when the associated dimension is synchronized.</td>
</tr>
</tbody>
</table>

**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.

**HierarchyProperties**

The `HierarchyProperties` dimension is used in the `HierarchyProperties` control cube, along with the `Hierarchies` dimension, to assign named levels to the hierarchy levels of a TM1 dimension.

The `HierarchyProperties` dimension contains the following elements:

<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. The element name you enter here may filter the dimension when TM1 data is retrieved from an external application like IBM Cognos Report Studio. 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>

**Hold**

The `Hold` dimension is used in any `Hold_UserName_CubeName` cubes.

*UserName* is the name of the TM1 user who has applied holds against cube values, while *CubeName* is the name of the cube in which values are being held.

The `Hold` dimension contains the following elements:

<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 <code>CubeName</code> 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 <code>UserName</code> in the <code>CubeName</code> cube. A value of H indicates that <code>UserName</code> has applied a leaf hold to a cell. A value of C indicates that <code>UserName</code> has applied a consolidation hold against a cell.</td>
</tr>
</tbody>
</table>
The }LineNumber control dimension contains numeric elements 1 through 10,000.

The elements in this control dimension correspond to the line numbers in a TM1 rules (.rux) file. The }LineNumber control dimension is used exclusively in the }StatsByRule control cube, which stores information and statistics about your TM1 rules.

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:

- }StatsbyClient
- }StatsByCube
- }StatsByCubeByClient

The }PerfClients dimension differs from the }Clients dimension in that the }PerfClients dimension contains numeric elements, while the }Clients dimension contains string elements.

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.

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.

The }RuleStats control dimension contains elements that correspond to the information and statistics that are stored for rules on your TM1 server.

This control dimension is used exclusively in the }StatsByRule control cube, which stores information and statistics about the rules on your TM1 server.

The following elements are included in the }RuleStats control dimension.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Text</td>
<td>The beginning of the rule text, to help you identify the rule.</td>
</tr>
<tr>
<td>Total Run Count</td>
<td>The total number of time the rule has been run.</td>
</tr>
<tr>
<td>Min Time</td>
<td>The minimum amount of time taken for the rule to run, in milliseconds.</td>
</tr>
<tr>
<td>Max Time</td>
<td>The maximum amount of time taken for the rule to run, in milliseconds.</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Avg Time</td>
<td>The average rule run time, in milliseconds.</td>
</tr>
<tr>
<td>Total Time</td>
<td>The total amount of time taken by the rule, in milliseconds.</td>
</tr>
<tr>
<td>Last Run Time</td>
<td>The amount of time, in milliseconds, it took for the most recent execution of the rule.</td>
</tr>
</tbody>
</table>

**}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 }StatsByClient cube.</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>

**}StatsStatsByCube**

The }StatsStatsByCube dimension is used only in the }StatsByCube control cube.

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

<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 TurboIntegrator 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>
</tbody>
</table>
### Number of Populated String Cells
This metric counts all the populated string cells in a given cube. String cells are cells identified by at least one string element.

### Number of Stored Calculated Cells
This metric counts all stored calculated cells including rules, dimensional consolidations, and user-defined consolidations.

### Number of Stored Views
This metric measures the number of named views of a given cube.

### Total Memory Used
The total amount of memory used by a cube, measured in bytes.

---

#### StatsStatsByCubeByClient
The `StatsStatsByCubeByClient` dimension is used only in the `StatsByCubeByClient` control cube. The dimension contains the following elements:

<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.
Appendix D. Excel Events Handled by TM1

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>
This information was developed for products and services offered worldwide.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service. This document may describe products, services, or features that are not included in the Program or license entitlement that you have purchased.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
Attention: Licensing
3755 Riverside Dr.
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information here is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:
© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

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

This Software Offering does not use cookies or other technologies to collect personally identifiable information.
Index

Special Characters
-h 10
.cfg file 9, 11
.ini file 17
.rej file 13, 20, 40
.xlsx worksheet 125
CellSecurity_CubeName control cube 143
Chores control dimension 155
ChoreSecurity control cube 143
ClientProperties control cube 147
ClientProperties control dimension 155
Clients control dimension 155
ClientsGroups control cube 146
ConnectionProperties control cube 147
ConnectionProperties control dimension 155
Connections control dimension 156
CubeFunctions control dimension 156
CubeProperties control cube 148
CubeProperties control dimension 157
Cubes control dimension 158
CubeSecurity control cube 144
DimensionAttributes control cube 149
DimensionAttributes control dimension 158
DimensionProperties control cube 148
DimensionProperties control dimension 158
Dimensions control dimension 159
DimensionSecurity control cube 145
ElementAttributes_DimensionName control cube 149
ElementAttributes_DimensionName control dimension 159
ElementSecurity_DimensionName control cube 145
Groups control dimension 160
Hierarchies control dimension 160
HierarchyProperties control cube 149
HierarchyProperties control dimension 160
Hold control dimension 160
Hold_UserName_CubeName control cube 153
LineNumber control dimension 161
PerfClients control dimension 161
PerfCubes control dimension 161
Processes control dimension 161
ProcessSecurity control cube 146
RuleStats control dimension 161
StatsByClient 55
StatsByClient control cube 150
StatsByCube 55, 151
StatsByCube control cube 151
StatsByCubeByClient 55
StatsByCubeByClient control cube 151
StatsByRule control cube 152
StatsForServer 55
StatsForServer control cube 153
StatsStatsByClient control dimension 162
StatsStatsByCube control dimension 162
StatsStatsByCubeByClient control dimension 163
StatsStatsForServer control dimension 163
TimeIntervals control dimension 163

A
access
control 27
privileges, and replication 134
privileges, groups 29
privileges, Read 134, 138
privileges, synchronization 138
privileges, Write 138
accessibility
keyboard shortcuts 141
add a server 90
Add Counters 68
ADMIN group
assigning passwords 34
Admin Host
location 11
resetting 17
specifying multiple 12
Admin server
diagram 9
IP address 9
name 9
port number 9
protocol 9
status 11
UNIX 11
Windows 10
Windows Startup directory 10
Windows status 11
Adminhost 90
AdminHost parameter
specify location 11
administration
tm1web_config.xml 108
administrative groups 30
AIX 126
alerts 69–71, 74
Allow Export as Text 3
Applications Server 97
assigning users to groups 33
Audit log
configuring 51
copying data 54
details 54
exporting results 54
find feature 54
object events 50
queries 52
results 53
system-wide events 50
understanding events 50
updating with latest records 52
using 50
viewing 52
Audit log, overview 50
authentication
integrated login 27
authentication, LDAP 27

**B**

background mode 17
Basic 92
Block Access to Server Capability Assignment 3
browser view 25
bulk load mode 25

**C**
cancel TM1 Top 64
Capability assignments 3
cell-level security 29
chain configuration
server 132
change password 35, 107
chore
  synchronization 140
clearing passwords 34
client
  connections 23
  disconnecting 24
  messages 23
  query 41
client logging 49
command prompt
TM1 Top 63
commit TM1 Top 62
configuration parameters
TM1 Web 108
configure
  TM1 Top utility 56
  configure login page 114
control cube
  CellSecurity_CubeName 143
  ChoreSecurity 143
  ClientProperties 147
  ClientSecurity 144
  ClientsGroups 146
  ConnectionProperties 147
  CubeFunctions 148
  CubeProperties 148
  CubeSecurity 144
  DimensionAttributes 149
  DimensionProperties 148
  DimensionSecurity 145
  ElementAttributes_DimensionName 149
  ElementSecurity_DimensionName 149
  HierarchyProperties 149
  Hold_UserName_CubeName 153
  ProcessSecurity 146
  StatsByClient 150
  StatsByCube 151
  StatsByCubeByClient 151
  StatsByRule 152
  StatsForServer 153
client and group administration 146
object attribute and property 147
performance monitoring 150
security 143
control dimension (continued)
  Clients 155
  ConnectionProperties 155
  Connections 156
  CubeFunctions 156
  CubeProperties 157
  Cubes 158
  DimensionAttributes 158
  DimensionProperties 158
  Dimensions 159
  ElementAttributes_DimensionName 159
  Groups 160
  Hierarchies 160
  HierarchyProperties 160
  Hold 160
  LineNumber 161
  PerfClients 161
  PerfCubes 161
  Processes 161
  RuleStats 161
  StatsStatsByClient 162
  StatsStatsByCube 162
  StatsStatsByCubeByClient 163
  StatsStatsForServer 163
  TimeIntervals 163
controlling object security 27
converting .xls 125
converting worksheets 126
copy
dimension 137
Counter Name 66
cube
  enabling and disabling logging 21, 41
  mirror 132
  replicating 131, 137
  security 29
  source 132, 136–139
  specifying replication information 137
  synchronizing 131, 138
Cube Dependancy 127
Cube Viewer
  page size 122
Cube Viewer, export sheets 122
CubeviewerStringWrap 122

data
  backup 20, 40
  data directory
    restricting access 34
Data Reservation
  capabilities 3
data, recovery 20, 40
DataAdmin group 31
Date and Time 48
DEBUG
  Admin server message severity 38
  logging system message severity 43
  TM1 Web message severity 123
Dependency in Cubes 127
dimension
  local 137
  rename 137
dimension (continued)
security 29
specifying replication information 137
dimension, copy 137
DisableSanding 3
disconnecting clients 24
Display user names 56, 91
Do Not Copy Rule 138
download configuration 102
download logs 102

E
editing tm1web_config.xml 108
element security 29
e-mail alerts 69–71, 74
End Time 41
Enhanced 92
ERROR
Admin server message severity 38
logging system message severity 43
TM1 Web message severity 123
exit TM1 Top 63
export
maximum sheets 122

F
FATAL logging system message severity 43
Filtering 101
fonts
Microsoft Excel default 126

G
garbage memory 24
group
adding 29, 32
admin 30, 34
multiple 33

H
health status 91
heartbeat interval 9, 10

I
IBM Cognos Performance Management Hub 81, 82
idle
TM1 Top 62
INFO
Admin server message severity 38
logging system message severity 43
TM1 Web message severity 123
integrated login 27
IX-lock 61

K
keyboard shortcuts
accessibility 141

L
LDAP
authentication 27
local dimension 137
local server 11, 18
lock modes 61
log
backing out records 22, 42
file 13, 21, 41
file, removing 22, 42
file, sample 20, 40
Tm1s.log file 40
viewing 21, 41
Log to disk 101, 102
Log_to_Disk 102
log, message 43
LogAppend 56
Logback 69–71, 74
logging
disabling 21, 41
sample properties file 44
system message severity 43
transactions 20, 40
Logging 100
Logging in 90
logging, enabling 21, 41
login page configuring 114
LogPeriod 56
LogReleaseLineCount 22

M
MaskUserNameInServerTools parameter 56, 91
MaxBackupIndex 46
MaxFileSize 46
memory
garbage 24
management 24
memory usage graph 94
MemorySize 46
message
clients 23
log 43
text 48
Message Level 48
Message log
configuring 45
properties file 44
severity Levels 43
TM1 loggers 44
using 43
metadata 13
Microsoft Excel
default font 126
mirror cube
definition 132
dimension 137
naming 137
rules 137, 138
specify information 137
synchronization 138, 139
monitoring chores in TM1 Operations Console 94
MTQ
MTQ (continued)
overview 2
multi-threaded queries, See MTQ

N
navigation tree
views node 121
numeric parameters 99

O
object security 27
operation group 90
Operations Console 97
out-of-process server 131
override dimension defaults 137

P
parameters, tm1web_config.xml 108
passwords
assigning 34
change 107
for the ADMIN ID 17
passwords, admin group 34
passwords, change 35
passwords, clearing 34
passwords, setting 34
performance counters 65
performance monitoring
}StatsByClient 150
}StatsByCube 151
}StatsForServer 153
control cubes 55, 150
Performance Statistics 93
Persisten feaders 6, 7
Personal Workspace Writeback Mode 3
PMHub, See IBM Cognos Performance Management Hub
privilege
read access 134, 138
write access 138

R
R-lock 61
read access privilege 134, 138
remote server
Admin host 11
and replication 132
application 14
connecting 17
diagram 13
disconnecting 18
memory management 24
operations 13
refreshing servers 17
replication 132
source server 132, 136
starting 15
target server 132, 136
Windows service 16
removing log files 22, 42
rename a dimension 137
replicate
a cube 132, 137
a dimension 137
Admin Host considerations 134
advantages 131
connection 132, 135
dimension name 137
mirror cube name 137
outcome 138
overview 131
process 136
required access privileges 134
server configurations 132
server relationships 132
setting up 134
source server 132
specifying rule information 138
specifying cube information 137
specifying dimension information 137
subset 138
target server 132
views 138
replication connection
creating 135
deleting 136
modifying 136
overview 132
rollback TM1 Top 62
rules 137, 138
RunningInBackground 17

S
sandbox
behavior in different clients 8
configuration parameters 8
delete sandboxes 7
files and folders 7
managing 7
memory usage 7
Sandbox Capability Assignment 3
sandbox queue 93
sandboxes, capability assignments 3
saving data 13
security
access privileges 29
adding groups 32
adding users 32
assigning passwords 34
change password 35
clearing passwords 34
cube-level 29
dimension-level 29
element-level 29
group 29, 30, 32–34
multiple groups 33
password security 34
setting passwords 34
TM1 data 34
TM1 objects 27
SecurityAdmin group 31
server
Admin 9
TM1 Top utility (continued)
  overview 56
  parameters 56
  thread processing states 62
TM1 Web
  administering 107
  configuration parameters 108
  homepage 116
  logging 123, 124
  startup parameters 119
  tm1web.log file 123
tm1admsrv-log.properties sample 38
Tm1admsrv.exe 10
tm1opsconsoleconfig.xml 102
Tm1p.ini file 17
tm1s_start_example 16
Tm1s.cfg file 9, 11
Tm1s.log
  properties 44
  removing old log files 22, 42
tm1srvstop.exe 15, 19
TM1StatusFilter 71
TM1ThresholdFilter 71
Tm1top.ini 56
tm1web_config.xml
  defined 108
  editing 114
  startup parameters 119
tm1xfer 83
top utility 56
transaction log
  query result 41
Transaction log 103
transaction log parameters
  Cubes 41
  Starting Time 41
transaction log parameters, Clients 41
transferring servers on Windows and UNIX 83
TurboIntegrator Process monitoring 93
U
UNIX 16
upload configuration 102
user
  adding 32
  user and group security 29
V
value 66
Verify admin access 91
verify TM1 Top 63
View log 101
view, Stargate 25
W
W-lock 61
WARN logging system message severity 43
Watchdog 104
watchdog rules 69–71
web.config 114
Windows
Windows (continued)
  service 10
Windows service 16
wrapping in cells 122
write access privilege 138