

IBM Planning Analytics  
Version 2 Release 0

*TM1 Web User Guide*



**Note**

Before you use this information and the product it supports, read the information in [“Notices” on page 163.](#)

**Product Information**

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

Licensed Materials - Property of IBM

Last updated: 2018-12-31

© **Copyright International Business Machines Corporation 2007, 2019.**

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

---

# Contents

<b>Introduction.....</b>	<b>vii</b>
<b>Chapter 1. What's new in TM1 Web.....</b>	<b>1</b>
2.0.6 - Feature updates, October 11, 2018.....	1
2.0.5 - Feature updates, June 25, 2018.....	1
2.0.3 - Feature updates, September 19, 2017.....	2
2.0.0 - Feature updates, December 16, 2016.....	3
<b>Chapter 2. TM1 Web overview.....</b>	<b>5</b>
Starting TM1 Web.....	5
Using TM1 Web.....	6
Data browsing and analysis tasks.....	6
Administrator tasks.....	6
Accessing TM1 Web from Apple iPad.....	6
Accessibility features.....	7
<b>Chapter 3. Working with websheets.....</b>	<b>9</b>
Websheet overview.....	9
Differences between websheets and Excel worksheets.....	9
Inherited Excel features in websheets.....	10
Viewing a websheet.....	12
Using the websheet toolbar.....	12
Active Forms in TM1 Web.....	13
Editing data in a websheet.....	13
Editing data in websheet cells.....	13
Using data spreading in a websheet.....	14
Excluding cells from data spreading.....	15
Excluding consolidations from data spreading.....	15
Working with relational data in websheets.....	16
Defining relational queries in Excel.....	16
Creating a parameterized query in Excel.....	18
Uploading a relational query to TM1 Web.....	19
Viewing relational data in TM1 Web.....	19
Changing websheet properties.....	20
Generating a report from a websheet.....	21
Websheet export limitations.....	23
<b>Chapter 4. Working in the TM1 Web Cube Viewer.....</b>	<b>25</b>
Opening a Cube View in TM1 Web .....	25
Using the TM1 Web Cube Viewer toolbar .....	26
Navigating pages.....	27
Saving data in a Cube View.....	28
Configuring a Cube View.....	29
Expanding and collapsing consolidations.....	29
Pivoting dimensions.....	29
Filtering a Cube View.....	30
Selecting elements from a subset.....	30
Drilling from a Cube View.....	31
Editing data in a Cube View.....	31
Editing data in Cube View cells.....	31
Using data spreading.....	32

Quick data entry commands.....	32
Entering data into consolidated cells on the Cube Viewer.....	35
Excluding cells from data spreading.....	35
Excluding consolidations from data spreading.....	35
Adding, viewing, and deleting comments in cells.....	36
Creating a new Cube View.....	36
Generating a report from a Cube View.....	37
Cube Viewer export limitation.....	38
<b>Chapter 5. Working with charts.....</b>	<b>39</b>
Changing the chart type.....	39
Drilling from a chart.....	39
<b>Chapter 6. Editing Subsets in TM1 Web.....</b>	<b>41</b>
Subset editing overview.....	41
Dynamic versus static Subsets.....	41
Opening the Subset editor.....	41
Editing with the Subset editor.....	41
Using the Subset editor toolbar.....	42
Displaying translated element names in the Cube Viewer.....	43
Moving elements.....	43
Moving consolidations.....	44
Keeping elements.....	44
Deleting elements.....	44
Filtering elements.....	45
Finding elements.....	46
Sorting elements.....	47
Expanding and collapsing consolidations.....	47
Inserting parents.....	49
Creating custom consolidations.....	49
Creating a custom consolidation from an existing Subset.....	49
Creating a custom consolidation from selected elements.....	50
<b>Chapter 7. Writeback modes and sandboxes.....</b>	<b>51</b>
Writeback modes.....	51
Setting the writeback mode.....	51
Understanding different toolbar options.....	52
Using direct writeback and named sandboxes.....	52
Using a personal workspace and named sandboxes.....	53
Personal workspace without named sandboxes.....	53
Direct writeback without sandboxes.....	54
Using a personal workspace or sandbox.....	54
Data values for leaf and consolidated cells in a sandbox.....	55
Resetting data values in a sandbox or personal workspace.....	55
Understanding cell coloring for changed data values.....	56
Committing changed data from a personal workspace or sandbox to base.....	56
Job queuing.....	57
Viewing the queue.....	58
Canceling a job in the queue.....	59
<b>Chapter 8. TM1 Web and scorecarding.....</b>	<b>61</b>
Scorecarding objects in TM1 Web.....	62
Metrics cubes in TM1 Web.....	62
Impact diagrams in TM1 Web.....	63
Strategy maps in TM1 Web.....	64
Custom diagrams in TM1 Web.....	65
Viewing metrics cubes in TM1 Web.....	66

Viewing impact diagrams in TM1 Web.....	67
Viewing strategy maps in TM1 Web.....	67
Viewing custom diagrams in TM1 Web.....	68
<b>Chapter 9. Administering IBM TM1 Web.....</b>	<b>69</b>
IBM TM1 Web overview.....	69
Changing Your password in TM1 Web.....	69
Configuring a proxy account for relational data connections.....	69
Modifying TM1 Web configuration parameters.....	70
TM1 Web configuration parameters.....	71
Editing the TM1 Web configuration file.....	77
Configuring the TM1 Web login page using AdminHostName and TM1ServerName parameters.....	77
Configuring a custom homepage for TM1 Web.....	78
Configuring TM1 Web startup and appearance settings.....	82
Changing the Cube Viewer page size.....	85
Setting the maximum number of sheets to export from a Cube Viewer.....	85
Wrapping string values in cube views.....	85
Setting the TM1 Web session timeout.....	86
Using TM1 Web logging.....	87
IBM TM1 Web log file.....	87
Message severity levels for TM1 Web logging.....	87
Configuring and enabling IBM TM1 Web logging.....	88
Viewing the TM1 Web log file.....	88
What if I use Microsoft Excel 2007 or earlier .xls worksheets?.....	89
Converting a .xls worksheet to .xlsx.....	89
Checking default font settings for non-Microsoft Windows web servers.....	90
<b>Appendix A. TM1 Web API.....</b>	<b>91</b>
TM1 Web API session login.....	91
Session token login.....	92
TM1 session ID login.....	94
Session and LoginDialog modules.....	94
TM1 Web URL API.....	97
TM1 Web URL API overview.....	97
Getting started with the TM1 Web URL API.....	98
TM1 Web URL API concepts.....	100
Displaying Websheet objects with the URL API.....	105
Displaying CubeViewer objects with the URL API.....	107
Upgrading older URL API projects to the new TM1 Web URL API.....	110
TM1 Web URL API parameter reference.....	111
TM1 Web JavaScript library.....	118
Required HTML <head> and <body> tags to use the JavaScript library.....	119
Configuring the AMD loader for the JavaScript Library.....	120
Loading websheet objects with the JavaScript library.....	122
Loading CubeViewer objects with the JavaScript library.....	123
JavaScript library callback functions.....	124
JavaScript library sample code for properties and methods.....	126
TM1 Web JavaScript library Workbook class.....	128
TM1 Web JavaScript library CubeViewer class.....	136
<b>Appendix B. Supported Microsoft Excel functions - TM1 Web.....</b>	<b>145</b>
Date and Time functions.....	145
Financial functions.....	145
Information functions.....	146
Logical functions.....	147
Lookup and reference functions.....	147
Math and trigonometric functions.....	148

Text and data functions.....	150
Statistical functions.....	151
<b>Appendix C. Unsupported Microsoft Excel functions - TM1 Web.....</b>	<b>155</b>
Database and list management functions.....	155
Date and time functions.....	155
Financial functions.....	156
Information functions.....	158
Lookup and reference functions.....	158
Math and trigonometric functions.....	159
Statistical functions.....	160
Text and data functions.....	161
<b>Notices.....</b>	<b>163</b>
<b>Index.....</b>	<b>167</b>

# Introduction

---

TM1<sup>®</sup> Web is a web-based software client that extends the analytical power of IBM<sup>®</sup> Planning Analytics.

With TM1 Web you can view, analyze, edit, and chart your IBM TM1 data in a Web browser. Administrators can also use TM1 Web to perform TM1 administration tasks.

**Note:** IBM Planning Analytics Workspace is the next-generation web-based interface for analyzing TM1 data, with ways to plan, create, and analyze your content. Planning Analytics Workspace combines the features and analytics of TM1 Web, TM1 Perspectives, and TM1 Architect. For more information, see the *Planning Analytics Workspace* documentation on IBM Knowledge Center.

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

## Finding information

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

## Samples disclaimer

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

## Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products.

TM1 Web includes accessibility features to help you perform tasks by using only a keyboard. These features include keyboard navigation and keyboard access to menus and dialog boxes that are related to websheets.

For more information, see [“Accessibility features”](#) on page 7.

## Forward-looking statements

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



---

# Chapter 1. What's new in TM1 Web

There are new features in IBM TM1 Web. For more information, see the *TM1 Web* documentation in [IBM Knowledge Center](#).

---

## 2.0.6 - Feature updates, October 11, 2018

IBM Planning Analytics Local release 2.0.6 and the cloud-only release of IBM Planning Analytics 2.0.6 includes the following features for TM1 Web.

### New parameter to configure session timeout for TM1 Web

In IBM Planning Analytics Local version 2.0.6, there is a new parameter in the `tm1web_config.xml` file called `HttpSessionTimeout`. This parameter defines the session timeout (in minutes) of the HTTP session for TM1 Web.

If the `HttpSessionTimeout` parameter is not specified (missing or blank), the value is less than 1 or not a numerical value, the default `session-timeout` that is defined in the `web.xml` file is used.

**Important:** As of IBM Planning Analytics Local version 2.0.6, you must not change the `session-timeout` value in the `web.xml` file.

For more information, see [“Setting the TM1 Web session timeout”](#) on page 86.

**Remember:** In IBM Planning Analytics Local version 2.0.6, you install a new version of the `tm1web_config.xml` file that is called `tm1web_config.xml.new` and your existing `tm1web_config.xml` file is preserved. To take advantage of fixes that are applied to this release, you must rename the `tm1web_config.xml.new` file to `tm1web_config.xml` and you must reapply any changes that you made to your previous configuration settings.

---

## 2.0.5 - Feature updates, June 25, 2018

IBM Planning Analytics Local release 2.0.5 and the cloud-only release of IBM Planning Analytics 2.0.5 includes the following features for TM1 Web.

### Changes to `web.xml` file

If you are upgrading IBM TM1 Web with this installation of IBM Planning Analytics Local release 2.0.5, you install a new version of the `web.xml` file called `web.xml.new` and your existing `web.xml` file is preserved.

To take advantage of fixes applied in this release, you must use the `web.xml.new` file and you must reapply any changes that you made to your previous configuration settings. In particular, restore your values for `session-timeout`.

1. Back up the existing `web.xml` file in `<PA_installation_location>/webapps/tm1web/WEB-INF`. For example, rename `web.xml` to `web.xml.old`. This step backs up your current settings.
2. Rename `web.xml.new` to `web.xml`. This step uses the new version of `web.xml` that is supplied with IBM Planning Analytics Local release 2.0.5.
3. Replace the entry for `session-timeout` in `web.xml` with the entry from `web.xml.old`. This step restores any changes that you made to this property previously.

For example:

```
<session-config>  
  <session-timeout>20</session-timeout>  
</session-config>
```

### **Exporting to PDF**

Exporting a worksheet to PDF uses the print area information. If you define a print area in a worksheet, only the data in that area will be exported to PDF. This behavior is the same as printing from Excel.

## **2.0.3 - Feature updates, September 19, 2017**

---

IBM Planning Analytics Local release 2.0.3 and the cloud-only release of IBM Planning Analytics 2.0.3 includes the following features for TM1 Web.

### **Display the current TM1 database label in TM1 Web**

The `TM1DatabaseLabel` parameter displays the TM1 database label in the banner beside the user name. For more information, see [TM1DatabaseLabel Parameter](#) and [TM1 Web Configuration Parameters](#).

### **Specify the maximum cell count of a workbook**

The `WorkbookMaxCellCount` parameter specifies the maximum cell count of a workbook as a number with no thousands separators. You can use `WorkbookMaxCellCount` to avoid issues opening workbooks with many cells.

For more information, see [TM1 Web Configuration Parameters](#).

### **Limit the number of cells that can be exported from worksheets**

The `ExportCellsThreshold` parameter specifies the maximum number of cells that an export of a worksheet 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.

For more information, see [TM1 Web Configuration Parameters](#).

### **Hide dimensions in the cube viewer**

The `CubeViewerHiddenDimensionsEnabled` parameter allows you to hide dimensions in the TM1 Web cube viewer.

For more information, see [TM1 Web Configuration Parameters](#).

### **Waterfall chart support**

TM1 Web supports excel-based Waterfall charts in worksheets. These charts were released in Microsoft Excel 2016.

## 2.0.0 - Feature updates, December 16, 2016

---

IBM Planning Analytics Local release 2.0.0 includes all features that were introduced in TM1 Web 10.3.0, which was introduced for IBM Planning Analytics on the cloud.

The following features were introduced in IBM Planning Analytics Local release 2.0.0. For more information about these features, see the *TM1 Web* documentation in [IBM Knowledge Center](#).

### Hierarchies in TM1 Web

TM1 websheets can display more than one hierarchy in a dimension.

**Note:** Hierarchies can be viewed in TM1 Web, however, you cannot create hierarchies in TM1 Web. You must create hierarchies in Planning Analytics Workspace. For more information, see Planning Analytics Workspace in [IBM Knowledge Center](#).

You can open hierarchies by using Quick Reports in IBM Planning Analytics for Microsoft Excel.

Quick Reports (formerly Flex Views) are published as live websheets. A live worksheet maintains its connection to the TM1 server. If the data on the server changes, the live worksheet reflects the change.

For more information about Quick Reports, see Planning Analytics for Microsoft Excel in [IBM Knowledge Center](#).

**Note:** Relative proportional spreading and relative percent adjustments are not supported in Quick Reports that are opened in TM1 Web.

### TM1 Web API enhancements

The TM1 Web API has the following new functionality:

- As of IBM Planning Analytics Local 2.0.0, it is no longer mandatory to use the version of Dojo that is provided with TM1 Web to load the TM1 Web JavaScript Library modules. TM1 Web now supports using the AMD loader from Dojo version 1.7 and later to load the JavaScript Library modules.
- The HTML <head> and <body> tags that are required to use the JavaScript library are simpler.
- The `tm1web/api/session/session` module in the JavaScript library allows users to log in, retrieve session information based on a session token, and destroy a session based on a session token.
- The `tm1web/api/session/LoginDialog` module in the JavaScript library allows users to display or destroy a login dialog box.
- The `tm1web/api/Workbook` class in the JavaScript library exposes execution information after an action button is executed. The `onActionButtonExecution` method API allows users to replace an existing Workbook or create a new one when an action button is clicked.
- The `tm1web/api/Workbook` class and the `tm1web/api/CubeViewer` class include subset and subsets set properties and methods.

For more information, see *TM1 Web API* in the *TM1 for Developers* documentation.

### Relational websheets

TM1 Web now allows you to view relational data on the same worksheet as TM1 data. By defining a relational query in an Excel file and then uploading the file to TM1 Web, you can view the results on the same worksheet or tab. This allows you to report on OLAP and relational data together.

For more information, see [Working with relational data in websheets](https://www.ibm.com/support/knowledgecenter/SSD29G_2.0.0/com.ibm.svg.ba.cognos.tm1_ug.2.0.0.doc/c_relational_data_websheets.html) ([https://www.ibm.com/support/knowledgecenter/SSD29G\\_2.0.0/com.ibm.svg.ba.cognos.tm1\\_ug.2.0.0.doc/c\\_relational\\_data\\_websheets.html](https://www.ibm.com/support/knowledgecenter/SSD29G_2.0.0/com.ibm.svg.ba.cognos.tm1_ug.2.0.0.doc/c_relational_data_websheets.html)) in IBM Knowledge Center.

## TM1 Web Accessibility

TM1 Web includes accessibility features to help you perform tasks by using only a keyboard. These features include keyboard navigation and keyboard access to menus and dialog boxes that are related to websheets.

- Context menus are accessed by using Shift+F10. The Up Arrow and Down Arrow keys select items from within the context menu.
- To expand or collapse a row in a worksheet, you can use the Space bar.
- To access the set selector, you can use the Space bar. The Tab key moves you between the search, the Arrow keys, and the tree. Up Arrow and Down Arrow keys move you between items in the tree. The Enter key selects the focused item in the tree.

**Note:** When you access the set selector, if you press Esc to exit after you make changes, you lose your focus on the cell that you originally launched from. You are focused on the main page.

## Support for Excel shapes in workbooks

Excel shapes, including basic shapes, arrows, banners, equation shapes, and lines, can be added to workbooks in TM1 Web. To see the list of supported and unsupported Excel shapes, see the [List of Microsoft Excel-supported functionality by menu in IBM TM1 Web release 10.2.2 and later](#).

## Single sign-on for TM1 Web

You can configure single sign-on for IBM TM1 Web by using Integrated Login (Kerberos) and the application server's security layer. Single sign-on enables HTTP users to log in only once to TM1 Web.

For more information, see *Configuring Integrated Login for TM1 Web using Kerberos and SPNEGO* in the *Planning Analytics Installation and Configuration* documentation.

## TM1 worksheet functions

The following worksheet functions are now available:

### TM1ELLIST

Returns a set of element values from a TM1 model by using a single formula.

### TM1GLOBALSANDBOX

Returns the current global active sandbox that was selected from the toolbar.

### TM1INFO

Returns information about the current TM1 version and client.

### TM1PRIMARYDB

Returns the primary TM1 server name that the user is authenticated through, even if the user is implicitly logged in to multiple TM1 servers.

---

## Chapter 2. TM1 Web overview

IBM TM1 Web extends the analytical power of IBM Planning Analytics by offering a number of tasks in a web browser.

With TM1 Web, you can:

- Analyze cube data.
- View and edit data in formatted Excel reports.
- Drill, pivot, select, and filter data.
- Build charts from cube data.
- Perform some IBM TM1 Server administration tasks.

**Note:** IBM Planning Analytics Workspace is the next-generation web-based interface for analyzing TM1 data, with ways to plan, create, and analyze your content. Planning Analytics Workspace combines the features and analytics of TM1 Web, TM1 Perspectives, and TM1 Architect. For more information, see the *Planning Analytics Workspace* documentation on IBM Knowledge Center.

---

### Starting TM1 Web

You can use the following steps to log in to IBM TM1 Web.

1. Start an internet browser.
2. Enter the URL provided by your TM1 Web administrator in the following format.

`http://machine_name:port_number/tm1web/`

Where:

**machine\_name**

The name of the Web server that is used to deliver TM1 Web pages.

**port\_number**

The port number of the Web server.

For example: `http://localhost:9510/tm1web/`

The TM1 Web login page opens.

3. Enter the login information.

**Admin host**

The name of the TM1 Admin Host you use to locate an active TM1 Server on your network.

**TM1 server**

The name of the TM1 Server that you want to access through TM1 Web. Click the down arrow to select one of the TM1 servers available on your network. Click **Refresh** to update the list of servers available on your network.

**Note:** If the AdminSvrSSLCertID parameter in the `TM1s.config` file is incorrectly configured, the server menu might be empty. For more information, see *Configuring the TM1 Server to use SSL* in the Planning Analytics Local Installation and Configuration documentation.

**User name**

Your user name on the selected TM1 Server.

**Password**

Your password on the selected TM1 Server.

4. Click **Login**.

The TM1 Web main page opens.

## Using TM1 Web

---

The main page of TM1 Web contains a Navigation pane on the left and a Content pane on the right.

### Navigation pane

The Navigation pane contains applications and views.

-  **Applications** Displays a list of applications that you can access through TM1 Web. These applications can contain shortcuts to TM1 websheets, cubes, and views.
-  **Views** Displays a list of cubes and views on the TM1 Server.

TM1 Web does not support the use of Back and Forward buttons from your browser. Use the controls that are offered in the Navigation pane to maintain consistent data views.

### Content pane

The Content pane displays the cube views and websheets that you open. Each object that you open displays on a separate tab.

## Data browsing and analysis tasks

---

TM1 Web provides tools for working with TM1 websheets, cube views, charts, and subsets.

For more information, see the following topics:

- [Working with websheets](#) describes how to view, edit, and export websheets.
- [Working in the TM1 Web Cube Viewer](#) describes how to view, edit, configure and export cube views, review and save data changes and create new views.
- [Working with Charts](#) provides details on using charts with TM1 Web Cube Views, changing chart properties, expanding and collapsing consolidations in a chart and drilling from a chart.
- [Editing Subsets in TM1 Web](#) describes how to use the TM1 Web Subset Editor to create and manage lists of elements that identify the data you want to analyze.

## Administrator tasks

---

As a TM1 Web administrator, you can perform administration and configuration tasks for the application.

For example:

- Change the password of the current user.
- Configure a custom homepage for TM1 Web.
- Modify TM1 Web configuration parameters.
- Use TM1 Web log files to monitor TM1 Web activity and errors.

For more information, see *Administering IBM TM1 Web* in the TM1 Operations documentation.

## Accessing TM1 Web from Apple iPad

---

You can access TM1 Web from an Apple iPad. TM1 Web is not supported on other mobile devices.

When you log in to TM1 Web from an iPad, the following limitations apply.

- The TM1 Web login screen must be used in portrait mode if you want to see the label for **Admin host** and the contents of the copyright.
- You cannot pinch to zoom in or out in Websheets on an iPad.
- Scrolling in the subset editor using touch gestures does not scroll in the subset editor but scrolls the TM1 Web page.
- The toolbar might not completely display in your Content pane and you cannot use touch gestures to scroll to the right of the toolbar. Collapse the Navigation pane to see the complete toolbar.
- You cannot resize dialog boxes with the resizing handle at the lower right of the dialog box by using touch gestures. Rotate your iPad to portrait or landscape mode to change the size of the dialog box.
- If you create a worksheet in IBM Planning Analytics for Microsoft Excel by using a template with a font that is not available on the iPad, the font might default to Times New Roman when you open the worksheet in TM1 Web.

## Accessibility features

---

People with limited vision can use screen-reader software, along with a digital speech synthesizer, to listen to what is displayed on the screen in TM1 Web. TM1 Web also includes accessibility features to help you perform tasks by using only a keyboard. These features include keyboard navigation and keyboard access to menus and dialog boxes that are related to worksheets.

- You can tab to all fields on the TM1 Web login page. The Down arrow reveals the **TM1 server** menu. You can use the Down and Up Arrow keys to navigate in the menu. You can use the Space bar to select from the menu. You can use the Space bar or Enter to refresh selections or select and clear check boxes. You can use the Space bar or Enter to click a button.
- On the banner, you can use the tab key to get to the **About** element. Press Enter to open the **About** dialog box. Tab to the **X** and use the Enter key to close the **About** dialog box. On the banner, you can also tab to **Help** or **Log Out**. Press Enter to open the help documentation in a new tab or press enter to log out.
- In the Navigation pane, you can use the Space bar to select a node, the Right arrow to expand nodes, the left arrow to collapse nodes, and the Up and Down arrows to move between nodes.
- On the toolbar, you can use the Right and Left arrow to move between elements. You can use the Space bar to click a button. When a toggle button is in focus, the space bar activates or deactivates the button. The copy and paste functions are not accessible from the toolbar; however, you can copy and paste from the context menu in a cell. All other actions on the toolbar can be triggered by pressing the Space bar.
- In the Content pane, context menus are accessed by using Shift+F10. The Up Arrow and Down Arrow keys select items from within the context menu.
- To expand or collapse a column or row in a worksheet, you can use the Space bar.
- To access the Subset editor, you can use the Arrow keys to navigate to a cell that has an element selector and then use the Space bar to open the element selector. The Tab key moves you between the search, the Arrow keys, and the tree. Tab to the bottom of the element selector to the Open subset editor icon  and then press the Space bar to open the Subset editor.
- In the Subset editor, Up Arrow and Down Arrow keys move you between items in the tree. The Enter key selects the focused item in the tree. You can also multi-select elements in the tree.
  1. Go to the first element that you want to select.
  2. Press Enter.
  3. Press the Down Arrow until you reach the last item that you want to select.
  4. Press Shift+Enter.
- In warning, error, and information dialog boxes, you can use the Tab key to navigate between buttons. You can use Insert+b to read the contents of the dialog box.

- In other dialog boxes, you can use Tab to navigate and Shift+Tab to navigate backwards. You can use Up and Down arrows to navigate between radio buttons and the Space bar to select and clear check boxes. You can use the Space bar or the Enter key to click buttons.

**Note:** When you access the set selector, if you press Esc to exit after you make changes, you lose your focus on the cell that you originally launched from. You are focused on the main page.

For more information about the commitment that IBM has to accessibility, see the [IBM Human Ability and Accessibility Center](http://www.ibm.com/able) (<http://www.ibm.com/able>).

---

## Chapter 3. Working with websheets

This section describes using websheets.

### Websheet overview

---

A websheet is a Microsoft Excel worksheet (.xls file) with IBM TM1 data that you can view in a web browser. By publishing an Excel worksheet from the IBM software to an application folder, other users can view your worksheet by using their Web browser.

With a websheet, you can perform the following tasks:

- Enter data in cells to which you have Write access. The IBM web client does not identify which cells are writable, so you must have some familiarity with your data to successfully enter data into the websheet. For more information, see [“Editing data in a websheet”](#) on page 13.
- Use data spreading to enter or modify many websheet values at the same time. Spreading is frequently used for scenario testing and what-if analysis during a budgeting or financial planning process.
- Drill to relational tables or other cubes. If the slice that you publish to the Web contains a cell with a defined drill-through rule, that drill function is available from your websheet.
- View Excel charts. If the slice you publish to the Web contains a chart, the chart appears in your websheet. If the slice from which you built the chart has a drill-through rule that is defined, you can drill through to related information from the websheet chart.
- Manipulate title element subsets in the Subset Editor.
- Display alternate hierarchies. You can view hierarchies in TM1 websheets, but you cannot create and manage hierarchies in a websheet.

The following Microsoft Excel ActiveX controls are supported in websheets:

- Check boxes
- Combo boxes
- List boxes
- Radio buttons
- Text boxes
- Labels

Diagonal borders are not supported in TM1 websheets.

### Differences between websheets and Excel worksheets

You might notice some differences between a TM1 websheet and an Excel worksheet.

#### Diagonal Borders

Diagonal borders are not supported in TM1 websheets.

#### Gridlines

If gridlines are enabled in an Excel worksheet, they also display in the associated TM1 websheet except for the following scenarios involving background color (cell shading):

- If gridlines are enabled in Excel and a background color is applied to the entire worksheet, the gridlines do not display in either Excel or the associated websheet.
- If gridlines are enabled in Excel and a background color is applied to only a range of cells in a worksheet, the gridlines for those cells are hidden in Excel but remain visible in the associated websheet.

## Inherited Excel features in websheets

A websheet inherits a subset of Excel features.

The following Excel features are inherited by websheets:

- Hide columns
- Conditional formatting
- Supported hyperlinks
- Freeze panes
- Cell protection (but not password protection)

### Hide columns

If you hide columns in your Excel worksheet, those columns are also hidden in the websheet. TM1 Web calculates the data cells whether they are visible in the websheet or not. If many hidden cells contain calculations, your websheet performance might be slower than you might expect.

### Conditional formatting

TM1 Web supports Excel conditional formatting, including color scale conditional formatting, and icon sets.

#### Icon set conditional formatting

With icon set conditional formatting, each icon represents a range of values. If color icons are used, the color of the icons represents a comparison of the values across the grid. Low values are red, medium values are yellow, and high values are green.

#### Color scale conditional formatting

With color scale conditional formatting, colors are shaded with gradations of two or three colors according to the value across the grid.

**Note:** When you use icon set or color scale conditional formatting in an Active Form, place the formatting on one row at a time, otherwise your results might not make sense.

#### *Using conditional formatting of repeated nested rows*

Labels of repeated nested rows in IBM TM1 Web are left blank. In Microsoft Excel, the text font for the repeated label matches the background color, and therefore cannot be seen. In TM1 Web websheets, the repeated cells are merged.

To see the hidden text in a worksheet, you can change the color of the font by using a conditional format on the outer rows to override the default setting. This method to override hidden text works both in Excel and in TM1 Web websheets:

### Procedure

1. Select **Conditional formatting > New rule**.
2. Select **Use a formula to determine which cells to format**.
3. In the field **Format values where this formula is true** type  $=1=1$
4. Click **Format** and select how you want the hidden text to be viewed.

### Hyperlinks

A subset of Microsoft Excel hyperlinks work in websheets.

- Another cell in the current workbook
- Named range that is defined in the current workbook
- Bookmark in the current workbook
- URL to an FTP or website
- Another Excel workbook. The target workbook can either be a file on your network or a file that is uploaded to the TM1 server.

If the target workbook is a file on your network, the hyperlink must contain the full network path to the target file that uses the Universal Naming Convention (UNC) format:

```
\\ComputerName\SharedFolder\FileName
```

For example:

```
\\system123\MyReports\hyperlink_target.xls
```

If the hyperlink points to a file uploaded to the TM1 server, the link must use the TM1 assigned name for the uploaded file. For more information, see the TM1 Developer documentation.

### **Freeze panes**

If you freeze panes in your Excel worksheet, the worksheet inherits the frozen panes. When you scroll vertically or horizontally in the worksheet, the frozen rows or columns remain visible.

If you scroll vertically in this worksheet, the rows in the frozen pane remain in place, while the lower portion of the worksheet scrolls.

### **Using ClearType to enhance display and rendering of worksheets**

To enhance the display of worksheets, especially ones that include a combination of frozen and unfrozen panes with wrapped text within cells, check with your administrator about installing the Microsoft ClearType Tuner.

This tool helps TM1 Web maintain the same row height between frozen and unfrozen panes in worksheets. For more information, see the section about administering TM1 Web in the *TM1 Operations* documentation.

### **String measurement for wide columns in TM1 Web**

StringMeasurement is a web.config parameter that determines the way the contents of a worksheet cell are adjusted to fit in columns.

When a column's width results in a cell that is smaller than its contents can display, the content is adjusted to that cell based on the StringMeasurement setting and the type of cell.

In all settings when the content is adjusted, digits are replaced with the '#' characters such that the number would not be mistakenly read as a different number.

If a disproportionately small amount of content is shown in your worksheet cells for the space available, you can use the legacy calculation by setting StringMeasurement=0 in the web.config file.

If too much content is shown in your worksheet for the space available, possibly causing misalignment, use the 1 - 3 settings, depending on the type of cell.

<b>String Measurement</b>	<b>Result</b>
0	Determines where to truncates string and number type cell content as it did before version 9.5.1.
1	String cell measurement uses the newer calculation.
2	Number cell measurement uses the newer calculation.
3	Both number and string content use the newer calculation.

### **Cell and password protection**

TM1 worksheets support cell protection that uses the Protect Sheet feature in Microsoft Excel, but do not support password protection. You can use Excel's Protect Sheet feature to protect your worksheet from data entry without entering a password.

Because a worksheet is a Web browser version of an Excel workbook, the integrity and layout of the workbook cannot be changed when the worksheet is accessed using a Web browser in TM1 Web. This type of access means that password protection is not typically needed in a worksheet.

## Viewing a worksheet

---

Any Excel worksheet that exists in a TM1 application is automatically available in TM1 Web.

### Procedure

1. From the browser, click an application in the left navigation pane.

The worksheets in the application appear as links in the list. Applications can contain references to various objects, such as cubes, dimensions, subsets, and views. Applications in TM1 Web display shortcuts to only worksheets, cubes, and views.

2. Click a worksheet link.

The worksheet appears in the browser.

If your administrator enabled the display of translated names on your TM1 server, then cubes, dimensions, elements, and attributes display in your local language as determined by the language setting of your Web browser. If translation is not enabled, object names appear as they were originally created on the TM1 server. In worksheets, only elements that are returned by SUBNM or TM1RptRow functions are translated. All other element and object names in worksheets display as originally created.

### What to do next

For more information about creating and managing applications, see the TM1 Developer documentation.

## Using the worksheet toolbar

---

The worksheet toolbar at the top of the TM1 Web page contains buttons for working with worksheets.

The following list describes the worksheet toolbar buttons.

### **Actions Menu**

Provides access to common worksheet tasks such as closing and exporting.

#### **Save data changes**

Saves changes to the currently selected worksheet.

#### **Export**

Exports the current worksheet to a Microsoft Excel slice, an Excel snapshot, or an Adobe PDF file.

#### **Reset data**

Clears all the changed data values that you entered up to that point in a sandbox. Resets all the data values back to the current values in the base data.

#### **Close**

Closes the currently selected worksheet.

#### **Close others**

Closes all worksheets, except the currently selected worksheet.

#### **Close all**

Closes all worksheets.

#### **Commit**

Sends the worksheet data modifications to the TM1 server.

### **Recalculate**

If you edited any data values in the worksheet, this option sends the data modifications to the TM1 server and then updates the data in the worksheet.

If you did not edit any data values in the worksheet, this option retrieves the current values from the TM1 server and updates the data in the worksheet.

### **Rebuild current sheet**

Rebuilds the current worksheet, including any Active Forms present in the worksheet.

### **Rebuild current book**

Rebuilds all worksheets/tabs in the current book, including any Active Forms present.

### **Autofit selected column width**

Adjusts the width of the currently selected column.

### **Sandbox**

Creates or deletes a sandbox. For more information, see [“Using a personal workspace or sandbox” on page 54.](#)

## Active Forms in TM1 Web

---

If a Microsoft Excel worksheet containing an Active Form is added to TM1 applications, the Active Form can be accessed in TM1 Web through the corresponding worksheet.

The following two new buttons are on the worksheet toolbar to simplify working with Active Forms.

Button	Name	Purpose
	Rebuild	Rebuilds the Active Form according to the form definition in the TM1RPTVIEW function.
	Column Resize	Widens the column in a worksheet to display all cell data. Select the column and click the button.

**Note:** If an Excel worksheet contains multiple Active Forms that originate from more than one TM1 Server, your username/password combination must be identical on all servers to successfully view the corresponding worksheet.

For example, if a worksheet contains one Active Form from ServerA and one Active Form from ServerB, the username/password combination you use to access ServerA must be identical to the username/password combination you use to access ServerB to successfully view the Active Forms in a single worksheet. If your username/password combination is not identical on all TM1 Servers represented in a worksheet, the worksheet will display incomplete data.

## Editing data in a worksheet

---

You can edit data in a worksheet by entering and editing values directly in the leaf cells of a worksheet or by using data spreading to distribute numeric values in a worksheet.

### Editing data in worksheet cells

You can edit data in the leaf cells of a worksheet if you have Write access to those cells. The TM1 Web client does not identify which cells are writable, so you must have some familiarity with your data to successfully enter data into the worksheet.

#### Procedure

1. Edit a value in a cell in one of the following ways.

### Edit the value

Double-click a value in a cell. TM1 Web displays the current value in the cell with a flashing cursor. The flashing cursor indicates that you can selectively edit the existing value by using the left and right arrow keys on your keyboard to position the cursor within the value. You can also use the Backspace and Delete keys to remove single numbers from the value.

### Replace the value

Single-click a value in a cell. TM1 Web displays the current value in the cell as highlighted, which indicates that the cell is in Edit mode. You can then type directly over the existing value in the cell, replacing it completely. You can also paste values into cells.

#### Note:

- When you paste a negative value, the value must be formatted with a leading minus sign, such as -1234.
- When you paste values with decimal places, you must be aware of the local browser settings for formatting. In some cases, pasting values with cell formatting for decimal places is not supported. If you use a number format in Excel, a number such as "123456.7" is copied as "123 456,7 " with a trailing space. A number parser might interpret the trailing space as a thousand separator in certain locales (for example, "fr") and might not allow the value to be pasted.
- Pasting values that are enclosed in parentheses is not supported in TM1 Web.

### Pick a new date value

If a cell is formatted as a date, double-click the cell then use the calendar to select a new date. Double-clicking also puts the cell in edit mode, so you can alternatively enter a new date directly in the cell.

Formatting is determined by the format attributes that are applied to the elements that identify a cell. For more information, see *Element Attributes* in TM1 Developer documentation.

When you type a value into a cell that has **Wrap Text** enabled, the row height expands as required to fit the new value. If a cell has **Wrap Text** enabled, but is merged into other rows/columns or has a custom height set on the row, the row height does not expand.

2. After you enter a new value, press **Enter** or click another cell.

The new number displays in bold and italic, which indicates that the value in this cell is new. You must submit the data changes to the TM1 server for the change you made to persist.

**Important:** If you log out of TM1 Web without submitting the new value, the change you made are lost.

3. Review your data changes.

If you are working in a sandbox, data changes display in a different color until the changes are committed.

4. Click **Commit**  on the websheet toolbar to save the changes to the server.

After you submit the changes, the websheet displays the updated values in a normal font, indicating that you saved the changes.

## Using data spreading in a websheet

You can use data spreading to enter or edit numeric data in a websheet with a predefined distribution method, called a data spread method.

For example, you can evenly distribute a value across a range of cells or increment all values in a range of cells by a percentage.

**Note:** TM1 Web saves the spread values to either the copy of an uploaded Excel file on the TM1 server or to the original location of an attached Excel file, depending on how the file was added to TM1 Web. You do not need to submit the data after TM1 Web completes the spread.

### Procedure

1. To spread data in a websheet, right-click on a cell and select **Data spread**.

2. From the Spreading menu, select any data spread method.

## Excluding cells from data spreading

You can apply a hold to cells to prevent those cells from being affected by data spreading.

You can still edit held cells. The holds apply only to the user who initiates the feature; other users can edit held cells.

### Apply a hold to a single cell or range

You can apply a hold to a single cell or range of cells.

#### Procedure

1. Select the cell or range.
2. Right-click the cell or range.
3. Click **Holds > Hold leaves**.

Each held cell displays a red triangle in the lower left corner as a visual indication that you applied a hold to that cell or range. When you log out, TM1 Web releases all holds.

### Release a hold on a single cell or range

You can release a hold on a single cell or range of cells.

#### Procedure

1. Select the cell or range of cells.
2. Right-click the cell or range.
3. Click **Holds > Release leaf holds**.

The released cells can accept values from data spreading operations.

**Note:** To release all holds you applied in a worksheet, right-click any cell in the worksheet and click **Holds > Release all holds**.

## Excluding consolidations from data spreading

You can hold the value of a consolidation constant while you adjust the underlying leaf values.

For example, you might want to hold a value constant while you change the values of the leaves to perform a what-if analysis.

When you apply a consolidation hold and change the value of its leaf elements, TM1 Web applies proportional spreading to the remaining leaf values so that the consolidation value remains unchanged.

### Apply a consolidation hold to a single cell or range

You can apply a consolidation hold to a single cell or range of cells.

#### Procedure

1. Select the cell or range.
2. Right-click the cell or range.
3. Click **Holds > Hold consolidate**.

Each held consolidation displays a red triangle in the lower left corner of a cell as a visual indication that you applied a hold to that cell or range. When you log out, TM1 Web releases all holds.

### Release a consolidation hold on a single cell or range

You can release a consolidation hold on a single cell or range of cells.

## Procedure

1. Select the cell or range of cells.
2. Right-click the cell or range.
3. Click **Holds > Release consolidate**.

The consolidated value now can reflect any changes that you make to the underlying leaf values.

**Note:** To release all holds you applied in a websheet, right-click any cell in the websheet and click **Holds > Release all holds**.

## Working with relational data in websheets

---

You can view relational data, formatted the way you want, on the same websheet as TM1 data.

One way to access a relational data source in TM1 is to use TurboIntegrator to extract relational data. You can then drill through to view the results. For more information, see the *TM1 TurboIntegrator* documentation.

The TurboIntegrator method has two limitations:

- In TM1 Web, the results appear on another websheet. In Excel, the results appear on another tab. This prevents you from including both OLAP and relational data in the same report.
- The results appear as a black and white table and cannot be formatted.

A second method allows you to view relational data without running any TI processes. You do this by defining a relational query in an Excel file and then uploading the file to TM1 Web.

- With this method, the results appear on the same websheet or tab, allowing you to report on OLAP and relational data together.
- You can format the report both in Excel and in TM1 Web.

## Defining relational queries in Excel

Connect to Microsoft SQL, IBM Db2®, and Oracle databases in Excel to define queries that will run in TM1 Web.

Before you can run a relational query in TM1 Web, you must author the query in Microsoft Excel.

For more information about querying a relational database using Microsoft Excel, see the documentation that was supplied with the Excel software.

### Creating a query of MS SQL data

**Note:** You do not need to install MS SQL Server OLE drivers; they are already installed with Microsoft Office.

1. In Microsoft Excel, go to the **Data** tab > **Get External Data > From Other Sources > From SQL Server**.
2. Enter the URL for the MS SQL Server database and then enter the user name and password.
3. Select a database and then select a table from the list.
4. Create the query.
  - a. Click **Properties > Definition** tab.
  - b. Change the command type to SQL.
  - c. Enter SQL commands in the **Command text** box.
  - d. Optionally, add parameters to your query.

For more information, see [“Creating a parameterized query in Excel” on page 18](#).

**Note:** You cannot validate the SQL query while you are creating it.

5. Optionally, modify and format the data in Excel.

**Note:** Most formatting will be retained when you upload the file to TM1 Web. However, table formatting is not retained.

6. Save the Excel worksheet.

### Creating a query of Db2 data

**Important:** Before you can connect to an IBM Db2 database from Excel, you must install the latest Db2 OLE drivers. For more information, see the [IBM Support Portal](http://www.ibm.com/support/entry/portal/support) (<http://www.ibm.com/support/entry/portal/support>).

1. In Microsoft Excel, go to the **Data** tab > **Get External Data** > **From Other Sources** > **From Data Connection Wizard**.
2. Click **Other/Advanced** and then click **Next**.
3. Select the Db2 OLE driver that you previously installed and then click **Next**.
4. Select **Direct server connection**.
5. Enter the server name and ODBC port number as follows:  
*server\_name:ODBC\_port\_number*
6. Select a database from the list and then enter a user name and password.
7. Select a table from the list and re-enter your user name and password, if required.
8. Create the query.
  - a. Click **Properties** > **Definition** tab.
  - b. Change the command type to SQL.
  - c. Enter SQL commands in the **Command text** box.
  - d. Optionally, add parameters to your query.

For more information, see [“Creating a parameterized query in Excel” on page 18](#).

**Note:** You cannot validate the SQL query while you are creating it.

9. Optionally, modify and format the data in Excel.

**Note:** Most formatting will be retained when you upload the file to TM1 Web. However, table formatting is not retained.

10. Save the Excel worksheet.

### Creating a query of Oracle data

**Important:** Before you can connect to an Oracle database from Excel, you must install the latest Oracle OLE drivers. For more information, see the [Oracle web site](http://www.oracle.com) (<http://www.oracle.com>).

1. In Microsoft Excel, go to the **Data** tab > **Get External Data** > **From Other Sources** > **From Data Connection Wizard**.
2. Click **Other/Advanced** and then click **Next**.
3. Select the Oracle OLE driver that you previously installed and then click **Next**.
4. Select **Direct server connection**.
5. Enter the server name, ODBC port number, and net service ID as follows:

*server\_name:ODBC\_port\_number/net\_service\_id*

**Note:** An error message may appear that ends with the following line:

IO Error: Invalid connection string format, a valid format is: "host:port:sid"

Despite this error message, the original connection was successful and the only syntax that will work is *host:port/sid*.

6. Select a database from the list and then enter a user name and password.

7. Select a table from the list and re-enter your user name and password, if required.
8. Create the query.
  - a. Click **Properties** > **Definition** tab.
  - b. Change the command type to SQL.
  - c. Enter SQL commands in the **Command text** box.
  - d. Optionally, add parameters to your query.

For more information, see [“Creating a parameterized query in Excel”](#) on page 18.

**Note:** You cannot validate the SQL query while you are creating it.

9. Optionally, modify and format the data in Excel.

**Note:** Most formatting will be retained when you upload the file to TM1 Web. However, table formatting is not retained.

10. Save the Excel worksheet.

You can now work with your relational data in TM1 Web. For more information, see [“Uploading a relational query to TM1 Web”](#) on page 19.

## Creating a parameterized query in Excel

You can create a parameterized query in Excel that can later be run from TM1 Web.

For more information about parameterized queries in Excel, see the Microsoft Excel documentation.

### Procedure

1. Follow the steps in the following web page:

[http://msdn.microsoft.com/en-us/office2010developertrainingcourse\\_vbalab\\_topic3.aspx](http://msdn.microsoft.com/en-us/office2010developertrainingcourse_vbalab_topic3.aspx) ([http://msdn.microsoft.com/en-us/office2010developertrainingcourse\\_vbalab\\_topic3.aspx](http://msdn.microsoft.com/en-us/office2010developertrainingcourse_vbalab_topic3.aspx)).

2. If the link above does not work, go to the [Microsoft web site](http://www.microsoft.com) (<http://www.microsoft.com>) and enter the search string "Creating a parameterized query in Excel".

#### **Db2 Note 1:**

When you apply a parameterized query in Db2, the following error may appear:

Function sequence error

To work around this issue, see the following technote:

<http://www.ibm.com/support/docview.wss?uid=swg21628120> (<http://www.ibm.com/support/docview.wss?uid=swg21628120>).

#### **Db2 Note 2:**

If you try to connect to a Db2 database using Microsoft Query, the OLE driver's connection string does not contain the hostname. An error message then appears saying that the connection failed. To fix this issue, you must add the following argument to the connection string:

```
location=<hostname>;
```

In addition, you must cancel out of any attempts that Excel makes to reconnect to the database, as that will reset the connection string.

## Uploading a relational query to TM1 Web

Upload relational queries to TM1 Web to query relational data in real time and view it on the same worksheet as TM1 data.

### Before you begin

Before you can upload a relational query to TM1 Web, you must create the query in Microsoft Excel and save the data as a worksheet. For more information, see [“Defining relational queries in Excel” on page 16.](#)

### Procedure

1. In TM1 Perspectives, open the Excel worksheet that contains the relational query.
2. On the **TM1** tab, click **Standard > Connect**.
3. In the **Server ID** field, select an application.
4. Enter the TM1 user id and password and click **OK**.
5. Click **Standard > Upload**.
6. Select the TM1 application folder where you want to upload the file and click **OK**.
7. Start TM1 Web by typing the following URL:

`http://tm1_server_name:9510/tm1web`

8. Expand the application folder and click the refresh button.

The relational worksheet appears in the folder.

## Viewing relational data in TM1 Web

You can view relational data in TM1 Web by running queries that were created in Microsoft Excel.

**Note:** You can format the data in TM1, However, you cannot write back to the relational database.

### Procedure

1. Start TM1 Web by typing the following URL:

`http://tm1_server_name:9510/tm1web`

2. Expand the application folder that contains the uploaded Excel worksheet.
3. Open the worksheet.

- a. Before the TM1 Web server displays the worksheet, it checks the RDBMS account information against the proxy account information that was defined by your administrator.

If this information matches, the worksheet is displayed in your TM1 Web window.

- b. If proxy account information is not found that matches the RDBMS account information, the TM1 Web server checks the RDBMS account information against your TM1 user name and password (Integrated Security mode 1 only).

If this information matches, the worksheet is displayed in your TM1 Web window.

- c. If neither the proxy account nor your TM1 account match the RDBMS account information, you are prompted to enter a user name and password that matches that of the relational data source.

If you enter a user name and password that matches that of the relational data source, the worksheet is displayed in your TM1 Web window.

**Note:** After three failed attempts, an error message appears. If this occurs, talk to your administrator to get a valid username and password.

## Changing websheet properties

---

Websheet properties determine how an Excel file displays and behaves when viewed as a websheet in TM1 Web.

All users can view websheet properties, but you must have Write access to an Excel file within an application to edit the websheet properties.

**Note:** You can manage websheet properties only by using Server Explorer, which is the user interface where you add Excel files to TM1 applications. The ability to manage websheet properties is not available directly in TM1 Web.

### Procedure

1. In the Tree pane of Server Explorer, locate the TM1 application that contains the Excel file for the corresponding websheet.

**Note:** You can access Server Explorer from IBM TM1 Perspectives or TM1 Architect.

2. Right-click the Excel file and click **Properties**.

The TM1 Web Properties dialog box opens, with two tabs:

- General
- Display Properties

3. If necessary, click the **General** tab to change the general properties.

#### TM1 Admin Hosts

Shows one or more admin hosts to which your server was registered when you generated an Excel slice. You can be connected to one or more admin hosts, and specify more than one admin host. Delimit each entry in the list with a semicolon.

#### Allow Write Back from Web

Allows users to modify TM1 data by entering values in the websheet. Disable this option to make the websheet read-only.

#### Print Properties

Sets a limit on the number of pages users can print from this websheet. The system default is 100. You can set this number to any value that is appropriate for this websheet. For example, to set the maximum number of pages users can print to 110, in the **Print Properties** section, enter **110** in the **Limit Number of Sheets to** box.

4. Click the **Display Properties** tab to change the display properties.

#### Display Title Element Selectors

Enable this option to display the Subset Editor buttons for title dimensions in the websheet. When this option is enabled, you can use the Display Selector option to selectively show/hide the Subset Editor button for individual title dimensions.

Clear this option to hide the Subset Editor buttons for all title dimensions in the websheet.

#### Title Dimensions

The Title Dimensions grid lists all title dimensions in the websheet. The grid has the following columns:

- **Dimension** - The name of the title dimension.
- **Address** - The cell address of the title dimension in the websheet.
- **Display Selector** - When the Display Title Element Selectors option is enabled, you can selectively show or hide the Subset Editor button for a title dimension in the websheet.

To show the Subset Editor button for a title dimension, select the corresponding check box in the Display Selector column.

To hide the Subset Editor button for a title dimension, clear the corresponding check box in the Display Selector column.

## Generating a report from a worksheet

---

You can generate reports in TM1 Web with worksheets and Cube Viewer.

### Worksheet

Select the title dimension subsets to include in the report. Optionally, you can define a print area of the worksheet that you want to export.

### Cube Viewer

Select the title dimension subsets and the number of rows to include in the report. For more information, see [“Generating a report from a Cube View”](#) on page 37.

**Note:** If your installation of TM1 Web is configured to run without Microsoft Excel on the Web server, some limitations might apply when you export worksheets. For more information, see [“Worksheet export limitations”](#) on page 23.

### Procedure

1. Click **Export** .
2. Select an export format for the report.



#### Slice to Excel

Excel documents that retain a link to the TM1 server by using functions. When you connect to the server that the slice is associated with, the slice displays the current cube values.



#### Snapshot to Excel

Excel documents that contain numeric values that reflect cube values at the moment the export occurred. Because snapshots do not retain a link to the TM1 server, the values are static, representing a snapshot of cube values at the moment of export.



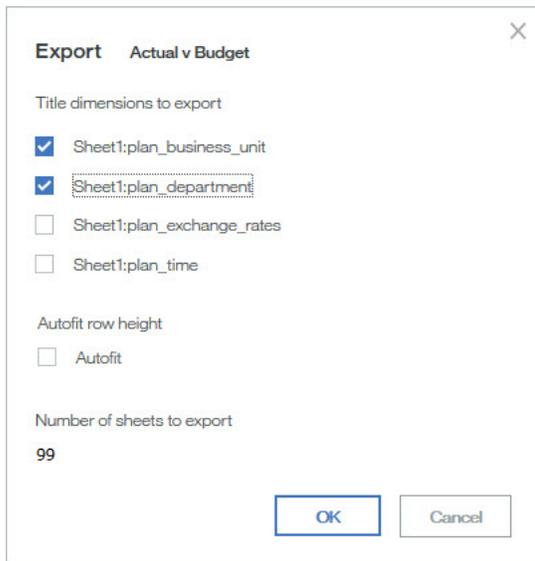
#### Export to PDF

PDF documents that display cube values at the moment the export occurred.

The worksheet Export dialog box opens. The dialog box reports the number of elements in each title dimension subset.

3. Select the title dimensions that you want to include in the report.

When you select dimensions, the dialog box indicates the number of sheets that will be generated. In the following example, where the plan\_business\_unit and plan\_department title dimensions are selected, the report will generate 99 sheets (9 elements x 11 elements).



**Note:** TM1 Web determines the number of elements for each title dimension by the number of elements in the current title dimension subset. If you edit a title dimension subset, the number of elements for the title dimension changes.

4. Click **OK** in the Export dialog box to create the report.

TM1 Web generates report sheets (or pages, for a PDF) by cycling through the selected title dimensions in the order they appear in the Export dialog box. In the example, TM1 Web generates the sheets as follows:

- For any title dimension not selected in the Export dialog box, TM1 Web uses the current title element in the worksheet in all report sheets. In the example, the model dimension is not selected, so TM1 Web uses the current title element in all report sheets.
- TM1 Web begins generating sheets by using the first element from the current subset of the actvsbud title dimension.
- Keeping the actvsbud title element constant, TM1 Web then generates sheets by cycling through all elements of the current subset of the region title dimension.
- TM1 Web generates sheets using the second element from the actvsbud title dimension subset.
- Keeping the second element from the actvsbud title dimension subset constant, TM1 Web generates sheets by again cycling through all elements of the current subset of the region title dimension.
- Finally, keeping the third element from the actvsbud title dimension subset constant, TM1 Web again generates sheets by cycling through all elements of the current subset of the region title dimension.

After TM1 Web generates all sheets, you can open or save the report.

5. Do one of the following steps:

- Click **Open** to open the report in a new browser window.
- Click **Save** to save the report to your hard disk.

**Note:** By default, exporting a slice or snapshot report to Excel displays the report in a web browser window. For more information on configuring your computer to open reports into the full, stand-alone version of Excel, see the Microsoft support website.

If you want to use TM1 functionality with a slice that you export to Excel, you must open the slice in the stand-alone version of Excel and have a local version of IBM TM1 Perspectives that is installed on your computer.

If you are experiencing problems exporting Excel or PDF files from TM1 Web, and TM1 Web is running on a WAN (wide area network) server, you might need to reconfigure the security settings in Internet Explorer. For more information, see the TM1 Operation documentation.

## **Websheet export limitations**

When Microsoft Excel is not present on the TM1 Web server, some limitations apply to exporting a websheet.

### **Slice to Excel/Snapshot to Excel**

- OLE controls in the websheet are converted to images
- Layout might be inconsistent between the websheet and the resulting Excel worksheet/workbook.
- Headers and footers in the worksheet are not exported
- Form control states are not updated or displayed in the resulting worksheet

### **Export to PDF**

- Exporting a websheet to PDF uses the print area information. If you define a print area in a worksheet, only the data in that area will be exported to PDF. This behavior is the same as printing from Excel.
- Images in the websheet are not exported
- Charts in the websheet are exported to a separate page in the PDF file
- OLE and form controls are not exported
- Headers and footers are not exported



# Chapter 4. Working in the TM1 Web Cube Viewer

This section describes working with a cube in TM1 Web.

## Opening a Cube View in TM1 Web

Follow these steps to open a cube view in TM1 Web.

### Procedure

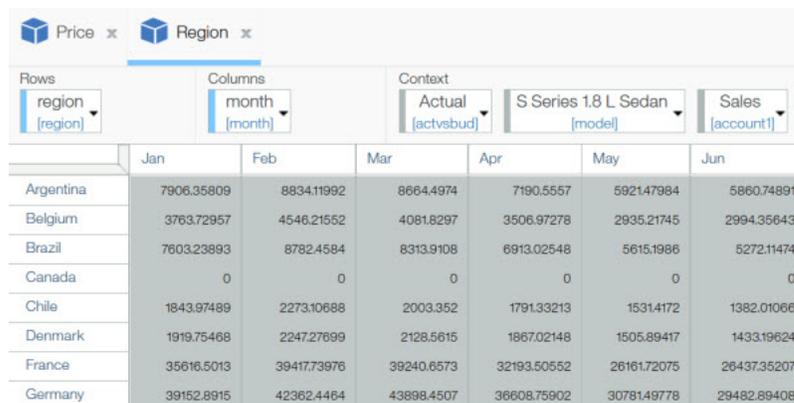
1. Log in to TM1 Web.
2. Open the Views node in the left Navigation pane.  
All cubes to which you have access appear in alphabetical order.
3. Click the **Expand** icon next to any cube to display the views available through TM1 Web.
4. Click a view in the list.

The view opens in the Content pane on the right. The Cube Viewer toolbar displays directly above the view.

5. Click another view in the Navigation pane.

The view opens in the Content pane and two View tabs appear above the Cube Viewer toolbar. Each View tab contains the name of an open view. The current view tab displays a border, indicating that the view is visible in the content pane.

The following example shows two view tabs: Price and Region. In this example, the Region tab displays with a border, indicating that the Region view is displayed in the Content pane.



	Jan	Feb	Mar	Apr	May	Jun
Argentina	7906.35809	8834.11992	8664.4974	7190.5557	5921.47984	5860.74891
Belgium	3763.72957	4546.21552	4081.8297	3506.97278	2935.21745	2994.35643
Brazil	7603.23893	8782.4584	8313.9108	6913.02548	5615.1986	5272.11474
Canada	0	0	0	0	0	0
Chile	1843.97489	2273.10688	2003.352	1791.33213	1531.4172	1382.01066
Denmark	1919.75468	2247.27699	2128.5615	1867.02148	1505.89417	1433.19624
France	35616.5013	39417.73976	39240.6573	32193.50552	26161.72075	26437.35207
Germany	39152.8915	42362.4464	43898.4507	36608.75902	30781.49778	29482.89408

Each time you open a view from the Navigation pane, the Cube Viewer displays a corresponding View tab above the Cube Viewer toolbar. When you open multiple views, the View tabs are organized horizontally along a single row with a set of arrow buttons that scroll left and right through the open tabs.

The following example shows multiple view tabs, with Budget Input Detailed as the current view tab.

	- Q1-2004	Jan-2004	Feb-2004	Mar-2004	- Q2-2004	Apr-
Sales	938,285	315,513	311,041	311,731	943,378	
Other Revenue	181,021	59,428	61,985	59,608	181,039	
<b>+ Revenue</b>	<b>1,119,306</b>	<b>374,941</b>	<b>373,027</b>	<b>371,339</b>	<b>1,124,417</b>	3
Direct Cost	154,925	37,737	52,376	64,813	142,410	
Other Costs	106,605	43,106	34,801	28,698	139,359	

6. Use the View tabs to display and close views:

- Click any **View** tab to display the corresponding view in the Content pane.
- Click **Close** on a View tab to close the corresponding view.
- Click the **Scroll left** and **Scroll right** arrows in the View tab scrollbar to navigate through the open View tabs.

## Using the TM1 Web Cube Viewer toolbar

The TM1 Web Cube Viewer toolbar buttons provide shortcuts to commonly used commands.

The following list describes each button in the toolbar.

### Actions menu

Provides access to common Cube Viewer tasks such as saving, closing, and exporting.

#### Save data changes

Saves changes to your current data.

#### Save view

Saves the current view to the TM1 server.

#### Save as

Saves the current cube view with a new name.

#### Export

Exports Cube Viewer data in the following formats:

**Slice to Excel** - Exports Cube Viewer data and TM1 formulas (SUBNM and DBRW functions) to a new Excel spreadsheet. The spreadsheet maintains a connection with the TM1 server.

**Snapshot to Excel** - Exports only Cube Viewer data to a new Excel spreadsheet, excluding the TM1 server formulas (SUBNM and DBRW functions). The spreadsheet does not maintain a connection with the TM1 server.

**Export to PDF** - Exports the Cube Viewer data to a PDF file. You must install a PostScript printer during the TM1 Web installation for the Export to PDF option to work. For more information, see the Planning Analytics Installation and Configuration documentation.

For more information on generating reports from a TM1 Web Cube Viewer, see [“Generating a report from a Cube View”](#) on page 37.

#### Reset data

Clears all the changed data values that you entered up to that point in a sandbox. Resets all the data values back to the current values in the base data.

### **Reset view**

Reloads the visual appearance of the Cube Viewer to the last saved arrangement of title dimensions.

### **Close**

Closes the currently selected cube view.

### **Close others**

Closes all cube views, except the currently selected view.

### **Close all**

Closes all cube views.

### **Commit**

Sends the changes you make to data in the Cube Viewer to the TM1 server.

### **Recalculate**

Updates the Cube Viewer configuration and recalculates data in the view. If you have edited any cells, all edits are automatically submitted to the TM1 server.

### **Auto calculation**

With the Auto Calculation option turned off, TM1 Web does not automatically recalculate the Cube Viewer when the view configuration changes.

For example, if you edit a row subset or move a dimension from the titles to the columns, these changes are not immediately displayed in the Cube Viewer; you must click the Recalculate button to see your changes.

When the Auto Calculation option is turned on, TM1 Web automatically recalculates the Cube Viewer when the view configuration changes.

### **Suppress zeros**

There are three options to suppress zeros:

- Suppresses zeros in rows and columns
- Suppresses zeros in rows
- Suppresses zeros in columns

### **View chart**

Displays the Cube Viewer data in a chart format.

### **View chart and grid**

Displays the Cube Viewer data in both grid and chart formats.

### **View grid**

Displays the Cube Viewer data in a grid format.

### **Chart Properties**

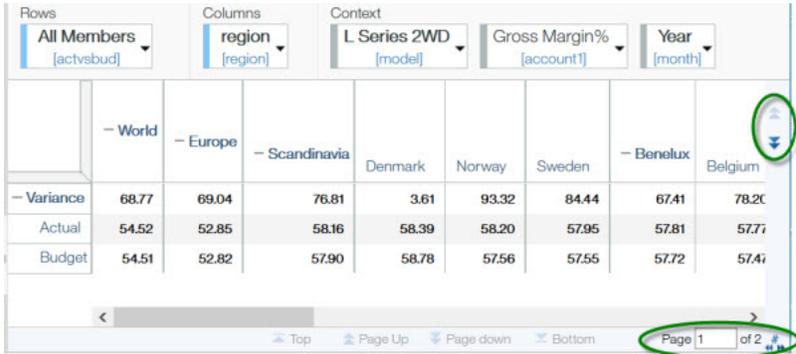
Displays options for selecting chart type or Scorecarding metric diagrams.

## Navigating pages

---

You can move from one part of a large cube view to another by navigating the pages.

A Paging toolbar is provided with navigation buttons and a Page indicator. In the cube view, the visible portion of the grid is the first of seven pages.



The following table contains the Paging toolbar buttons and indicator with their descriptions.

Button or Indicator	Name	Description
	Display Pages	Displays the CubeView PageLayout dialog box with a layout of all pages. Click a page to navigate to a specific page.
	Previous Page (Rows)	Shows the previous page of rows.
	Next Page (Rows)	Show the next page of rows.
	Next Page (Columns)	Shows the next page of columns.
	Previous Page (Columns)	Shows the previous page of columns.
	Page Indicator	Displays the current page and the total number of pages of cells in the view.

## Saving data in a Cube View

You can save data changes from TM1 Web to the server.

### Procedure

1. Click **Save view**  or **Recalculate**  to save the changes to the data.

If you click **Save view**, TM1 Web displays a message that asks if you want to save the changes to the Cube Viewer data.

2. Click one of the following buttons:

- **Yes** - Submits the data changes to the server, recalculates the view, and returns to the Cube Viewer. If you changed the view configuration, the configuration is saved as well.
- **No** - Discards the data changes and returns to the Cube Viewer.
- **Cancel** - Returns to the Cube Viewer. The data changes remain visible in the Cube Viewer.

3. Click **Save data changes**  to save the changes.

## Configuring a Cube View

You can re-configure the Cube Viewer in various ways to arrive at a view that satisfies your reporting or analysis needs.

- Expand and collapse consolidations
- Pivot dimensions
- Filter view data
- Edit subsets
- Drill through to associated data

### Expanding and collapsing consolidations

You can click the control next to an element name to expand or collapse a consolidation in the Cube Viewer.

**Expand +** - A plus sign next to an element name identifies the element as a consolidation. To drill down on consolidations in a dimension and view the underlying detail, click the plus sign. The plus sign changes to a minus sign.

**Collapse -** - A minus sign next to an element name indicates an expanded consolidation. To roll up the leaf elements in a dimension, click the minus sign. The minus sign changes to a plus sign.

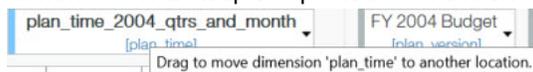
### Pivoting dimensions

You can pivot the dimensions in your Cube Viewer to change the presentation of cube data. To pivot dimensions, use the drag-and-drop operation.

- Drag a dimension to the column position.
- Drag a dimension to the row position.
- Drag a dimension to the title position.

When you drag a dimension to a new position, three possible options are available when you drop the dimension. The options vary by the position of your cursor. The following examples use dimensions named Dimension1 and Dimension2.

- When you drag Dimension1 and position your cursor in the center of Dimension2, dropping the dimension will swap the positions of the two dimensions.



- When you drag Dimension1 and position your cursor on the left side of Dimension2, Dimension1 is dropped immediately to the left of Dimension2.



- When you drag Dimension1 and position your cursor on the right side of Dimension2, Dimension1 is dropped immediately to the right of Dimension2.



If you drag a dimension and drop it immediately to the left or right of an existing column or row dimension, you can see more detail along the columns or rows of a view.

## Filtering a Cube View

You can filter data in a cube view that contains a single row dimension and one or more column dimensions.

When you have two or more dimensions along the columns, you can filter only from the innermost dimension, that is the dimension closest to the view grid.

### Procedure

1. Click the column element that contains the values that you want to filter.
2. Select a filter.
  - **Pre-defined filter** - Top 10, Bottom 10, Top 10 Percent, Bottom 10 Percent. The filter is immediately applied to the view.
  - **Advanced** - You can define a custom filter by setting filter parameters in the Filter dialog box, as described in the following steps.
3. Select a **Filter** type.

Filter Type	Description
TopCount	Filters the view to display only the largest n elements, where n is a number specified in the Value option.
BottomCount	Filters the view to display only the smallest n elements, where n is a number specified in the Value option.
TopSum	Filters the view to display only the largest elements whose sum is greater than or equal to n, where n is a number specified in the Value option.
BottomSum	Filters the view to display only the smallest elements whose sum is greater than or equal to n, where n is a number specified in the Value option.
TopPercent	Filters the view to display only the largest elements whose sum is greater than or equal to n, where n is a percentage of the dimension total specified in the Value option.
BottomPercent	Filters the view to display only the smallest elements whose sum is greater than or equal to n, where n is a percentage of the dimension total specified in the Value option.

4. Enter a numeric value in the **Value** box.
5. Select a **Sort** order to display the dimension elements in the Cube Viewer in ascending or descending order.
6. Click **OK**.

### Results

A small funnel icon displays next to the column element for which you created a filter.

**Note:** To remove a filter, click the column element for which you created the filter, and click **Remove Filter**.

## Selecting elements from a subset

You can select one or more elements from a subset and view the elements, along with associated data, in the Cube Viewer.

## Procedure

1. Click the **Open Subset Editor** ▾ button next to any subset.  
The Subset Editor window opens in your browser.
2. Select the element(s) you want to see in the Cube Viewer.
3. Click **OK**.

## Drilling from a Cube View

In Perspectives and Architect, you can set up drill processes and drill assignments to access related information in your cube views.

When these drill processes and rules are in place, they are available in TM1 Web. You can use these drill processes and rules to drill to another cube view or to a relational database.

## Procedure

1. To drill to another cube view, right-click a cell and click **Drill**.  
The target cube view containing information related to the cell opens.
2. To drill through from one cube view to another, right-click a cell and click **Drill**.  
The target Cube Viewer opens on a new tab.

## Editing data in a Cube View

---

You can edit data in the TM1 Web Cube Viewer.

### Editing data in Cube View cells

You can edit data in leaf cells if you have Write access to those cells.

Leaf cells appear with a white background in the Cube Viewer.

If you are working in a sandbox, you can save the sandbox to store your values across sessions. For more information, see [Chapter 7, "Writeback modes and sandboxes,"](#) on page 51.

## Procedure

1. Edit a value in a white cell in one of the following ways.

### Edit the value

Double-click a value in a white cell. TM1 Web displays the current value in the cell with a border, a white background, and a blinking cursor. This indicates that you can selectively edit the existing value by using the left and right arrow keys on your keyboard to position the cursor within the value. You can also use the Backspace and Delete keys to remove single numbers from the value.

### Replace the value

Single-click a value in a white cell. TM1 Web displays the current value in the cell as highlighted, which indicates that the cell is in Edit mode. You can then type directly over the existing value in the cell, replacing it completely. You can also paste values into cells.

### Note:

- When you paste a negative value, the value must be formatted with a leading minus sign, such as -1234.
- When you paste values with decimal places, you must be aware of the local browser settings for formatting. In some cases, pasting values with cell formatting for decimal places is not supported. If you use a number format in Excel, a number such as "123456.7" is copied as "123 456,7 " with a trailing space. A number parser might interpret the trailing space as a thousand separator in certain locales (for example, "fr") and might not allow the value to be pasted.
- Pasting values that are enclosed in parentheses is not supported in TM1 Web.

### Pick a new date value

If a cell is formatted as a date, double-click the cell then use the calendar to select a new date. Double-clicking also puts the cell in edit mode, so you can alternatively enter a new date directly in the cell.

Formatting is determined by the format attributes that are applied to the elements that identify a cell. For more information, see *Element Attributes* in TM1 Developer documentation.

2. After you enter a new value, press **Enter** or click another cell.

**Note:** When you enter a number into a consolidated cell in the web Cube Viewer, the value is proportionally spread across the consolidation. For example, if you enter **50** into a consolidated cell in the web cube viewer, the value is spread across the consolidation as if you had entered spreading code of 50p. This behavior occurs only in the web Cube Viewer. In Architect/Server Explorer Cube Viewer and in slices from Perspectives and websheets, you must enter the spreading code to get the value to spread proportionally across the consolidated cells.

The new number displays in bold and italic, which indicates there is a new value in this cell. You must submit the view to the server for the change you made to persist.

**Important:** If you log out of TM1 Web without submitting the new value, the change you made is lost.

3. Review your data changes.

If you are working in a sandbox, data changes display in a different color until the changes are committed.

4. Click **Commit**  on the Cube Viewer toolbar to save the changes to the server.

The Cube Viewer displays the updated values. All values appear in a normal font, indicating that you saved the changes.

## Using data spreading

You can use data spreading to enter or edit numeric data using a predefined distribution method, called a data spread method.

For example, you can evenly distribute a value across a range of cells, or increment all values in a range of cells by a percentage.

### Procedure

1. To spread data, right-click a cell and click **Data Spread**.
2. From the Spreading menu, select any data spread method.

**Note:** TM1 Web saves the spread values to the server. You do not need to submit the data after TM1 Web completes the spread.

## Quick data entry commands

Typing a data entry command in a cell performs an action on the cell value.

Data entry commands are processed when you press Enter. These commands only apply to the current grid.

These commands are not case-sensitive.

You can use commands across two dimensions, but not across pages.

The following table lists the quick data entry commands.

Command	Description	Action
K	Enters the value in thousands.	Example: 5K Enters 5,000

Command	Description	Action
M	Enters the value in millions.	Example: 10M Enters 10,000,000
Add, +	Adds a number to the cell value.	Example: Add50 Adds 50 from the cell value
Subtract, Sub, ~	Subtracts a number from the cell value. <b>Important:</b> A minus sign (-) is not permitted for subtract because this indicates a negative number.	Example: sub8 Subtracts 8 from the cell value
Percent, per	Multiplies the cell value by a number added as a percentage.	Example: per5 Gives 5% of the original cell value
Increase, Inc	Increases the cell value by a number added as a percentage.	
Decrease, Dec	Decreases the cell value by a number added as a percentage.	Example: decrease6 Decreases the cell value by 6%
GR	Grows cells by a percentage.	Example: GR>150:10 Increases the value by 10 percent starting with a value of 150.
Hold, Hol, H, HC	Holds the cell value from breakback calculations. HC holds the consolidated level.	
Release, Rel, RH, RC	Releases held cells.	
RA	Release all held cells.	

### Using shortcuts in different clients

There are shortcut keys available in the IBM TM1 Application Web client.

The following table shows the shortcut keys available in the IBM TM1 Application Web client and in TM1. Note that not all shortcuts available in IBM Planning Contributor are also available in TM1. See also the notes at the end of the table for important information about using shortcut keys.

Application Web	TM1
Add10	P+10
Sub10	P~10
Increase10	P%+10
Decrease10	P%~10
Percent10	P%10

Application Web	TM1
Add10> or >Add10	R+>10
Sub10> or >Sub10	R~>10
Increase10> or >Increase10	P%+>10
Decrease10> or <Decrease10	P%~>10
Percent10> or >Percent10	P%>10
>10	R>10
10>	R>10
>10K	R>10000
>10M	R>10000000
10Grow100Compound>	GR>10:100
10Grow100Linear>	GR>10:100
10Gro100Com>	GR>10:100
10Gro100Lin>	GR>10:100
10G100C>	GR>10:100
10G100L>	GR>10:100
10Grow100>	GR>10:100
1K	1000 (The number ending in K is multiplied by 1000 at the client end and returned to the server)
1M	1000000 (The number ending in M is multiplied by 1000000 at the client end and returned to the server)

- When a shortcut such as 10K is entered, the numbers are multiplied by 1000, or 1000000 at the client end and then the shortcut is converted to the equivalent spreadcode.
- The TM1 spreadcodes cannot be used in combination with Planning Contributor shortcuts. For example, P%Add10 or RPAdd10 are not allowed. Also, Planning Contributor shortcuts cannot be used in combination with TM1 shortcuts. For example, Add10Sub20 is an invalid entry.
- The Planning Contributor shortcuts of Multiply, Divide, Power and Reset are not available in TM1.
- All Grow commands whether Compound or Linear, are converted to the TM1 GR spreadcode command. GR command can only do a Linear Growth
- The direction of spread can be entered at the start or the end of the shortcut. Shortcut strings with the direction in the middle are invalid. For example, Add10> or >Add10 are correct, but Add>10 or Add1>0 are invalid.
- All shortcut codes are *not* case sensitive. For example, add10, Add10, or aDD10 produce the same result.

## Entering data into consolidated cells on the Cube Viewer

When you enter a number into a consolidated cell in the Cube Viewer, the value is proportionally spread across the consolidation.

For example, if you enter 50 into a consolidated cell in the Cube Viewer, the value is spread across the consolidation as if you had entered the spreading code of 50p. This behavior occurs only in the Cube Viewer. In the Architect/Server Explorer Cube Viewer and in slices from Perspectives and in websheets, you must enter the spreading code to get the value to spread proportionally across the consolidated cells.

## Excluding cells from data spreading

You can apply a hold to cells to prevent those cells from being affected by data spreading. You can still edit held cells.

The holds apply only to the user initiating the feature; other users can edit held cells.

### Apply a hold to a single cell or range

You can apply a hold to a single cell or range.

#### Procedure

1. Select the cell or range.
2. Right-click the cell or range.
3. Click **Holds > Hold leaves**.

#### Results

Each held cell displays a red triangle in the lower-left corner as a visual indication that you applied a hold to that cell or range. When you log off, all holds are released.

### Release a hold on a single cell or range

You can release a hold on a single cell or range.

#### Procedure

1. Select the cell or range of cells.
2. Right-click the cell or range.
3. Click **Holds > Release leaf holds**.

#### Results

The released cells can accept values from data spreading operations.

**Note:** To release all holds that you applied to all cubes, right-click any cell in any cube, click **Holds > Release all holds**.

## Excluding consolidations from data spreading

You can hold the value of a consolidation constant while adjusting the underlying leaf values. For example, when performing a what-if analysis you might want to hold a value constant while changing the values of the leaves.

When you apply a consolidation hold and change the value of its leaf elements, proportional spreading is applied to the remaining leaf values so that the consolidation value remains unchanged.

### Apply a consolidation hold to a single cell or range

You can apply a consolidation hold to a single cell or range.

#### Procedure

1. Select the cell or range.

2. Right-click the cell or range.
3. Click **Holds > Hold consolidate**.

### Results

Each held consolidation displays a red triangle in the lower-left corner of a cell as a visual indication that you applied a hold to that cell or range. When you log off, all holds are released.

### Release a consolidation hold on a single cell or range

You can release a consolidation hold on a single cell or range.

### Procedure

1. Select the cell or range of cells.
2. Right-click the cell or range.
3. Click **Holds > Release consolidate**.

### Results

The consolidated value can now reflect any changes that you make to the underlying leaf values.

**Note:** To release all holds that you applied to all cubes, right-click any cell in any cube, click **Holds > Release all holds**.

## Adding, viewing, and deleting comments in cells

You can add or view comment text in any cell in a cube view.

Comments that you attach to a cell in TM1 Web can be viewed in IBM Cognos Insight (Standalone mode) or in IBM Cognos Performance Modeler. To delete a comment, you can purge commentary from the Cognos TM1 Applications portal. For more information, see "Configuring commentary on applications" in the TM1 Performance Modeler documentation.

### Procedure

1. In TM1 Web, select the cell in which you want to add or view comment text.
2. To add a comment, follow these steps:
  - a) Right-click the cell and click **Add comment**.
  - b) Type the text for the comment.

**Tip:** A small red triangle appears in the corner of the cell to indicate that the cell contains comments that are attached to it.

3. If the cell already has comments, click **Browse comments**.

A table appears that lists each comment, along with the user name and the date that the comment was added.

## Creating a new Cube View

---

If the views for a cube do not satisfy your analysis requirements, you can create a new view.

### Procedure

1. Expand the Views node in the left navigation pane.
2. Double-click a cube name
  - If you have a private default view of the cube, TM1 Web displays it in the content pane.
  - If you do not have a private default view of the cube, but a public default view exists, TM1 Web displays the public view in the Content pane.

- If you have neither a private default view nor a public default view of the cube, TM1 Web displays the system default view in the Content pane. In the system default view, the last dimension in the cube definition is the column dimension; the next-to-last dimension in the cube definition is the row dimension; and other dimensions are context dimensions.

Or, expand the cube and click an existing view.

3. Modify the view to meet your requirements. See [“Configuring a Cube View” on page 29](#).
4. Click **Actions** > **Save as**.
5. Type a name for the view.
6. Decide whether you want to create a public or private view. A private view can be viewed only by you.
  - To create a private view, select the **Private** check box.
  - To create a public view, clear the **Private** check box.

**Note:** You must be the TM1 administrator or have Admin privileges on the cube to save a public view.

7. To save the view as the default view for the cube, click **Default**.
  - For example, if you select the **Private** check box and the **Default** check box, the view is saved as your private default view of the cube. The next time you double-click the cube, this is the view you will see.
  - If you clear the **Private** check box and select the **Default** check box, the view is saved as the default view of the cube for all users on the server. The next time a user double-clicks the cube, this is the view that they will see unless they have created their own private default view of the cube.

**Note:** You must be the TM1 administrator or have Admin privileges on the cube to save a public view.

8. Click **OK**.

**Important:** If you do not save the view, TM1 Web discards the view when you close the view or end your TM1 Web session

## Generating a report from a Cube View

---

You can generate briefing book-style reports in two ways:

- **Cube Viewer** - Select the title dimension subsets and the number of rows to include in the report.
- **Websheet** - Select the title dimension subsets to include in the report. For more information, see [Chapter 3, “Working with websheets,” on page 9](#).

**Note:** If your installation of TM1 Web is configured to run without Microsoft Excel on the Web server, some limitations may apply when exporting from a Cube Viewer. For more information, see [“Cube Viewer export limitation” on page 38](#).

### Procedure

1. Click **Export** .
2. Select an export format for the report:
  -  **Slice to Excel** - Excel documents that retain a link to the server through TM1 functions. When you open the slice and connect to the server with which the slice is associated, the slice displays the current cube values, provided you are running Excel with the Perspectives add-in enabled.
  -  **Snapshot to Excel** - Excel documents that contain numeric values reflecting the cube values at the moment the export occurred. Because snapshots do not retain a link to the server, the values are static, representing a snapshot of cube values at the moment of export.
  -  **Export to PDF** - PDF documents that display cube values at the moment the export occurred.

The Export dialog box opens.

3. Select the number of rows to export:

- **Export rows in current page** - Exports all rows in the current page.
  - **Export rows from beginning to current page** - Exports the first row in the first page through the last row in the current page.
  - **Export all rows in the view** - Exports all rows from all pages.
4. Select the title dimensions that you want to include in the report.
  5. Click **OK** to create the report.

The report sheets are generated and prompts you to either open or save the report.

6. Do one of the following:
  - Click **Open** to open the report in a new browser window.
  - Click **Save** to save the report to disk.

**Note:** By default, exporting a slice or snapshot report to Excel displays the report in a web browser window.

For more information on configuring your computer to open reports into the full, stand-alone version of Excel, see the Microsoft support web site.

Additionally, if you want to use TM1 functionality with a slice that you export to Excel, you must open the slice in the stand-alone version of Excel and have a local version of Perspectives or Client installed on your computer.

**Note:** If you are experiencing problems exporting Excel or PDF files and you are using a WAN (Wide Area Network) server, you may need to re-configure the security settings in Internet Explorer. For more information, see the "Administrating TM1 Web" section of the *Planning Analytics Installation and Configuration* documentation.

### **Cube Viewer export limitation**

When Microsoft Excel is not present on the TM1 Web server, and you export a Cube Viewer using either the Slice to Excel or Snapshot to Excel options, any charts present in the Cube Viewer are not exported to the resulting worksheet.

---

## Chapter 5. Working with charts

This section illustrates how to view a chart in TM1 Web.

### Procedure

1. Open a view.
2. Do one of the following to view a chart:
  - Click **View chart** to view cube data in chart format only.  
A column chart, the default chart type, is displayed.
  - Click **View chart and grid** to view cube data in both chart and grid format.  
A grid is displayed at the top, and a column chart, the default chart type, is displayed at the bottom.
  - Click **View grid** to view cube data in grid format only.

---

### Changing the chart type

You can change the chart type from the Chart Properties menu.

Follow the steps below to change the chart type.

### Procedure

1. On the toolbar, click **Chart properties > Chart type**.
2. Select one of the available chart types, such as **Point**, **Line**, **Column**, or **Pie**.

---

### Drilling from a chart

If your administrator has defined drill-through processes and rules for cube cells represented in a chart, you can drill through to associated data from the chart.

For more information on creating drill-through processes and rules, see the TM1 Developer documentation.

If a chart component is associated with a single source of associated data, the data immediately opens on a new View tab. If the chart component is associated with a multiple sources of associated data, you are prompted to select a single source.

For example, this section illustrates how to execute a drill.

### Procedure

1. Click **View chart** to display the chart.
2. Right-click a column in the chart and click **Drill through**.

If the cell is linked with two or more sources of associated data, the Drill dialog box opens, listing the data sources associated with the chart component.

3. Select the source you want to view and click **Select**.

### Results

The selected data opens on a new **View** tab.



---

## Chapter 6. Editing Subsets in TM1 Web

This section describes how to use the IBM TM1 Web Subset Editor to create and manage lists of elements that identify the data you want to analyze.

### Subset editing overview

---

The Subset Editor tool lets you define a subset for any dimension to limit the number of elements used in a view.

A dimension can have thousands of elements. It is unlikely, however, that any view will require all elements from all dimensions. In most cases, you should limit the elements used in a view to those that are required for a specific analysis of your data.

For best results, limit the number of elements that appear as title elements. That way, if you view the data over slower Internet connections, your data displays more efficiently.

### Dynamic versus static Subsets

When you open a dynamic subset in TM1 Web, a warning message displays informing you that the dynamic subset will be converted into a static subset: This Subset was created using an expression. Modifying this subset will delete the expression and convert the subset into a static subset.

After you make changes to the subset, and save the subset, TM1 Web replaces the dynamic subset with a static subset.

To edit a dynamic subset without converting it to a static subset, use the Server Explorer Subset Editor.

### Opening the Subset editor

---

You can open a Subset Editor from a worksheet or Cube Viewer.

#### Procedure

1. From a worksheet, click **Open subset editor**  at the far right end of a title dimension.
2. From a Cube Viewer, click **Open subset editor** ▼ at the far right end of a subset.

### Editing with the Subset editor

---

To perform editing tasks on a subset, use the Subset Editor.

#### Procedure

1. Click **Open subset editor** ▼ next to any dimension.  
The Subset opens.
2. Click **Open subset editor**  at the bottom of the Subset.

#### Results

The Subset Editor contains two panes.

#### Available elements (left pane)

Displays all the elements that are available to be added to your subset.

### Subset (right pane)

Displays only the actual members of the subset. When you save a subset, only the elements in the Subset pane are saved to the subset.

### Using the Subset editor toolbar

The editing tasks available in the Subset Editor are accessed from its toolbar buttons.

The following table describes the Subset Editor toolbar buttons:

Button	Name	Description
	Save subset	Saves only the elements that appear in the Subset list to the subset.
	Save subset as	Saves only the elements that appear in the Subset list to the subset with a different name.
	Reload subset	Reloads the original subset.
	Subset all	Displays all the elements in the parent dimension.
	Cut, Copy and Paste	Cuts, copies, and pastes the selected elements of a subset.
	Keep selected elements	Keeps elements that you select for the subset.
	Delete selected elements	Removes elements that you select from the subset.
	Filter subset	Allows you to select a group of elements in a subset that have related characteristics. You can filter elements in these ways: <ul style="list-style-type: none"><li>• Filter by Level</li><li>• Filter by Attribute</li><li>• Filter by Expression</li></ul>
	Sort subset	Lets you sort a subset in several ways: <ul style="list-style-type: none"><li>• Sort Ascending</li><li>• Sort Descending</li><li>• Sort Hierarchically</li><li>• Sort by Index Ascending</li><li>• Sort by Index Descending</li></ul>
	Tree expand	Expands the tree in several ways: <ul style="list-style-type: none"><li>• Drill Down Selected Consolidations - Expands the selected consolidation one level.</li><li>• Expand Selected Consolidations - Expands the selected consolidation, showing all descendents.</li><li>• Expand Tree Fully - Expands the entire hierarchy, showing all children of all parents.</li></ul>

Button	Name	Description
	Tree collapse	Collapses the tree in two ways: <ul style="list-style-type: none"> <li>• Collapse Selected Consolidations - Collapses the expanded consolidation, hiding all descendents.</li> <li>• Collapse Tree Fully - Collapses the entire hierarchy.</li> </ul>
	Insert parents of selected elements	Inserts the parent of the selected element immediately above that element in the hierarchy tree.
	Expand above	Displays consolidations at the bottom of the list of children, in both the Available Elements and Subset lists. The children of the consolidation expand above the consolidation.
	Create custom consolidation	Allows you to build consolidated elements on the fly when working with a view.  For more information, see <a href="#">“Creating custom consolidations”</a> on page 49.
	Find in subset	Enables you to search for elements in the current subset based on the search text you enter.

## Displaying translated element names in the Cube Viewer

When your model has been translated, as described in *Translating your model* in the TM1 Developer documentation, you can display translated element names in the cube viewer.

### Before you begin

Ensure that the language in which you want to view element names is set as the display language for your browser.

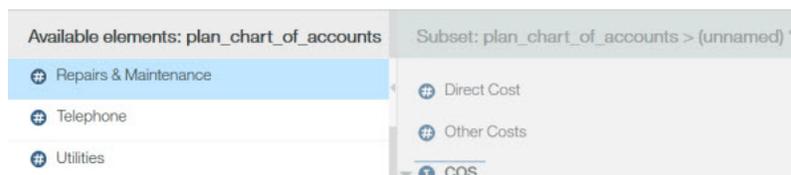
### Procedure

1. In the cube viewer, click the dimension for which you want to display translated element names. The current subset of the dimension opens in the **Subset editor**.
2. In the **Subset editor**, select **Caption** from the **Alias** list.
3. Click **OK**.
4. Close and reopen the view, saving any changes if prompted. The elements display in the language used by your web browser.

## Moving elements

You can move elements from the Available Elements pane to the Subset pane using a drag-and-drop operation.

In this example, if you click Repairs & Maintenance in the Available Elements pane, you could drag the element to under Other Costs in the Subset pane.



The line beneath the Other Costs element indicates that the Repairs & Maintenance element will appear beneath Other Costs.

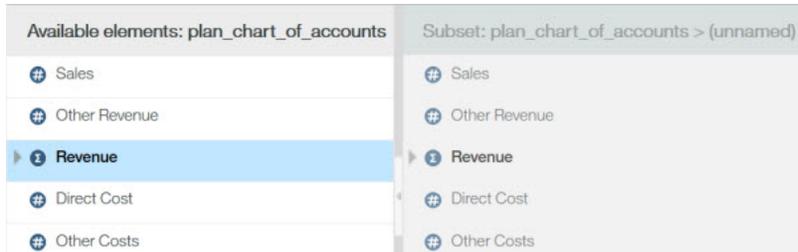
## Moving consolidations

You can move a consolidation from the Available Elements pane to the Subset pane using a drag-and-drop operation.

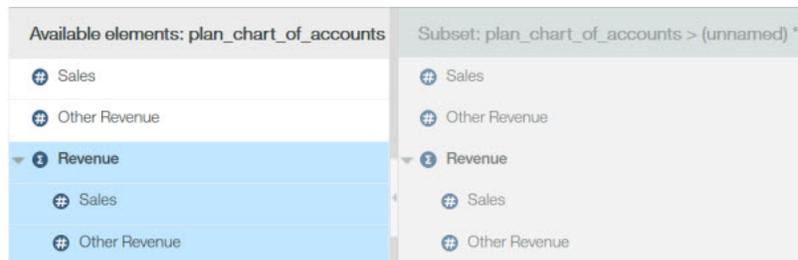
When you move a consolidated element, the children of the consolidation also move.

For this example, suppose you have a consolidation element named Revenue.

If you select Revenue, and drag it to the Subset pane, a collapsed consolidation is added to the Subset pane.



If you expand Revenue in the Available Elements pane, and select the consolidation and its children, you can drag the consolidation to the Subset pane. The expanded consolidation is added to the Subset pane.



In both of the examples, the Revenue consolidation and its children are added to the Subset list. However, the state of the consolidation in the Subset list reflects the way the consolidation displays. In the first example, Revenue displays as a collapsed consolidation. In the second example, Revenue displays as an expanded consolidation and its children will be visible.

## Keeping elements

You can reduce the list of elements in the Subset pane to only those elements you want to keep in your subset.

In this case all other elements are removed from the subset.

**Note:** You can reduce the size of the Available Elements list to narrow your search for elements to add to your subset, but this has no effect on the elements in the Subset list.

### Procedure

1. Select the elements that you want to keep in the Subset list.
2. Click **Keep selected element(s)**  .  
Only the elements that you selected to keep remain visible in the Subset list.
3. Click **Save subset**  to save the subset.

## Deleting elements

You can remove selected elements from the Subset pane.

## Procedure

1. Select one or more elements in the Subset pane.
2. Click **Delete selected element(s)** .

## Results

The selected elements are removed from the Subset pane. The removed elements still exist in the dimension.

**Note:** To display all subset elements that you removed, click **Subset all** .

## Filtering elements

You can filter elements in either the Available Elements pane or Subset pane.

Use these options:

- **Filter by attribute** - Displays only the elements that match an attribute that you specify.
- **Filter by level** - Displays only the elements that match a level in the element hierarchy.
- **Filter by expression** - Displays only the elements that match a pattern.

### Filtering by attribute

The Subset Editor lets you filter elements by attribute value.

## Procedure

1. Click **Filter subset**, , and click **Filter by attribute**.
2. In the **Select attribute** list, select an attribute.
3. In the **Select value to match** list, select a value.
4. Click **OK**.

## Results

All subset elements whose selected attribute matches this value remain in the element list. All subset elements whose selected attribute does not match the value are removed from the element list.

### Filtering by level

The Subset Editor lets you filter elements so that only elements belonging to one or more specified hierarchy levels remain.

Consider the following example of a three-level hierarchy.

In this example, you start with the subset shown in the figure, and then eliminate all elements from the subset except those at Level 1.



## Procedure

1. Click **Filter subset** , and click **Filter by level**.

2. Click a level in the list, and click **OK**.

For example, if you filtered by **Level 1**, the following level 1 subset elements remain in the Subset list:

- Revenue
- COS

### Filtering by expression

The Subset Editor lets you filter elements so that only elements matching a specified search pattern remain.

For example, suppose you have the following list of elements in either the Available Elements pane or Subset pane.

- Sales
- Other Revenue
- Direct Cost
- Other Costs
- Bank Charges
- Board of Directors
- Employee Relations
- Printing
- Seminars and Continuing Ed.
- Taxes and Licenses
- Office Expense
- Postage
- Rent

Now suppose you want to reduce this list to those elements that contain the word 'cost'.

### Procedure

1. Click **Filter subset**  and click **Filter by wildcard**.
2. Enter a pattern of alphanumeric characters in the **Enter expression** box.

You can use the following two wildcard characters in the **Enter expression** box.

- **Question mark (?)** - Placeholder for a single character
- **Asterisk (\*)** - Placeholder for one or more characters

To isolate all elements whose names contain the string pattern *cost*, type the expression ' cost ' in the dialog box that opens.

3. Click **OK**.

### Results

The element list is trimmed to include only those elements that match the pattern.

## Finding elements

You can search for elements in either the Available Elements pane or Subset pane by using the Find in Subset toolbar.

This feature performs a simple text search for elements that match a spelling pattern that you enter. This is especially useful when you want to find a specific element within a large list of elements.

**Note:** The Find in Subset feature does not support wildcard characters, such as the question mark (?) or asterisk (\*), in your search text. Instead, an asterisk (\*) wildcard character is inserted at the beginning and

end of the spelling pattern that you enter so that it searches for any occurrence of the pattern in the element list.

For example, if you enter the spelling pattern ost, this converts to \*ost\* and matches such as Cost and Boston are found.

### Procedure

1. Click **Find in subset**  or press Ctrl+F.

The Find in subset toolbar opens in the Subset editor.

2. Type a spelling pattern in the search box.

A spelling pattern can include one or more alphanumeric characters, but should not include wildcard characters.

The list of elements is searched as you type a spelling pattern.

- If one or more matching elements are found, the first matching element is located and highlighted in the list.
- If a matching element is not found, the search box temporarily displays a red background.

You can also start your search at any location within the element list by clicking on an element in that section of the list. The search begins from this new start point when you continue your search.

3. Click **Find next** or **Find previous** to navigate through the element list when more than one matching element is found.

You can also use the following keyboard commands to navigate:

- Press F3 or press Enter to find the next matching element.
- Press Shift+F3 or press Shift+Enter to find the previous element.

If a next or previous matching element is not found, the search box temporarily displays a red background, and the search cycles through the list again.

4. Click **Close the findbar**  to close the **Find in subset** toolbar.

## Sorting elements

You can sort all elements in either the Available elements pane or Subset pane.

### Procedure

To sort subset elements, click **Sort subset**  and select a sort option.

Sort option	Sort order
Sort ascending	Ascending order from A to Z, 0-9.
Sort descending	Descending order from Z to A, 9-0.
Sort hierarchically	All children appear beneath their parents.
Sort by index ascending	Dimension index, starting at 1.
Sort by index descending	Dimension index, starting at the highest index in the dimension.

## Expanding and collapsing consolidations

You can expand a consolidation in the Subset Editor to display the immediate children or all descendants of the consolidation.

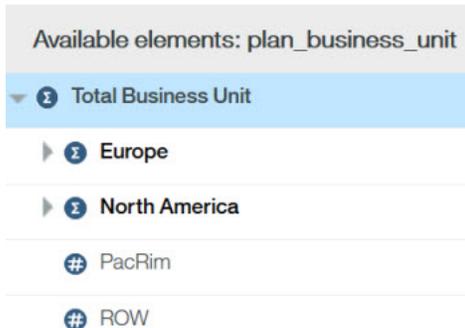
You can apply the following procedures to elements in either the Available Elements pane or the Subset pane of the Subset Editor.

### Expanding a consolidation

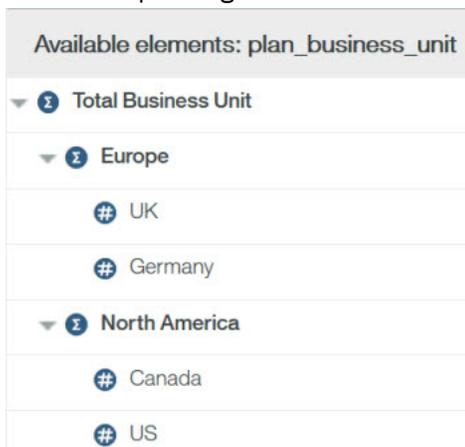
You can expand a consolidation.

#### Procedure

1. Select the consolidations you want to expand.
2. Click **Tree expand** .
3. Select one of the following:
  - Click **Drill down selected consolidations** to view the immediate children of a consolidation. The following figure shows the result of drilling down on the Total Business Unit consolidation. The immediate children of Total Business Unit display when you click **Drill down selected consolidation**.



- Click **Expand tree fully** to view all descendents of a consolidation. The following figure shows the result of expanding the Total Business Unit consolidation.



- Click **Expand tree fully** to view all descendents of all parents in the dimension hierarchy.

### Collapsing a consolidation

You can collapse expanded consolidations using either a selected consolidation or you can close all expanded consolidations in the subset.

#### Procedure

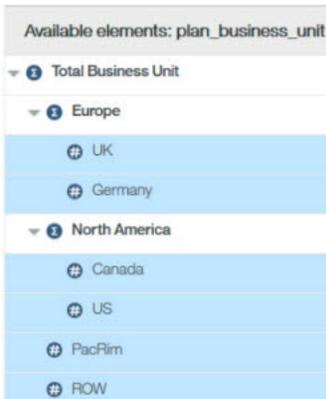
1. Select the expanded consolidations you want to collapse.
2. Click **Tree collapse** .
3. Click **Collapse selected consolidations**.

**Note:** To close all expanded consolidations in the subset, click **Tree collapse**,  and click **Collapse tree fully**.

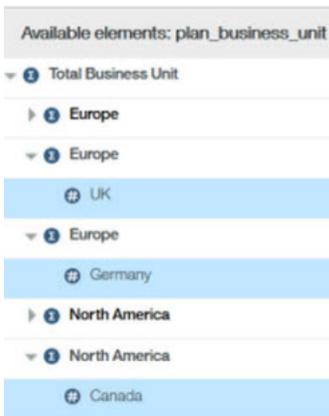
## Inserting parents

You can insert the immediate parent of a selected element directly above that element in the Subset Editor.

For example, consider the following example showing several leaf elements.



If you select all elements, and click **Insert parents of selected elements** , the immediate parents of all selected elements are inserted, as shown in the following example.



## Creating custom consolidations

When working with a view, you can create custom consolidations from existing subsets or from selected subset elements.

### Creating a custom consolidation from an existing Subset

You can create a custom consolidation by inserting an existing subset into the current subset.

The existing subset then becomes a custom consolidation within the current subset.

#### Procedure

1. Open the Subset Editor for a dimension.
2. Define a subset in the **Subset** pane.
3. Click **Create custom consolidation**  and click **Create consolidation from subset**.
4. Select the existing subset that you want to insert into the current subset as a custom consolidation.

The selected subset is inserted into the current subset as a custom consolidation.

5. If necessary, click **Save subset**  or **Save subset as**  to save the current subset.

6. Click **OK**.

### Results

The subset with the new custom consolidation opens.

## Creating a custom consolidation from selected elements

You can create a custom consolidation from selected elements in the Subset Editor.

### Procedure

1. Open the [Subset Editor](#) for a dimension.
2. In the Subset pane, select the elements that you want to include in the custom consolidation.
3. Click **Create custom consolidation**,  and click **Create consolidation from selected elements**.

You have now created a custom consolidation that contains the elements that you selected in step 2.

The custom consolidation the name **}ROLLUP\_#** is assigned, where # starts at zero and increases by one for each custom consolidation that you create during a server session.

4. Click **OK** to view the new custom consolidation.

---

## Chapter 7. Writeback modes and sandboxes

IBM TM1 offers different ways to work with data changes.

The Writeback mode in combination with the type of Sandbox determines how changes to the server data are managed. These different options allow the administrator to mix and match a variety of capabilities so that every installation and every usergroup can work in the way that is best for them. TM1 also offers [Job Queuing](#) to more efficiently process data change submissions to the server.

For more information, see [Using a Personal Workspace or Sandboxes](#).

---

### Writeback modes

In IBM TM1, you can hold changes in a private area so that you can decide manually when to write the data changes back to the server and thereby make your changes available to others. This private area is called a personal workspace or a sandbox, depending on the extent of its capabilities. When you commit the data changes that were in your private area to the base data, the changed values are written to the server.

If you prefer to work directly with the base data without a private workspace, you can choose a direct writeback method. Another option your administrator can offer is the ability to name and store data changes in a named sandbox.

When you work in a sandbox or personal workspace, TM1 uses a change in cell coloring to remind you when your data is not yet merged with the base. Once you commit the sandbox or personal workspace, the cell color is restored to black. See [Understanding cell coloring for changed data values](#) for more information.

Your Administrator assigns the capabilities for each usergroup. Since you could be a member of more than one group, your work space options can be different depending on your login, the client you use, and the combination of settings. Only Administrators have access to the Capability Assignments.

Ask your administrator for more information about how your system is designed to operate. See [Understanding different toolbar options](#) to learn how to determine your writeback mode and sandbox setting using the toolbar. For more information about Capability Assignments, see the *TM1 Operations* documentation.

### Setting the writeback mode

The Writeback Mode Capability determines how data is written back to the server. Writeback mode is determined by whether a user has the personal workspace capability on or off.

Description	Personal workspace mode Capability mode
Changes are made directly to the base.	Off
Changes are held in a temporary area and are manually written to the base using the Commit button or option. Cell coloring changes when data is changed but not yet committed. You can process using the Job Queue.	On

The Sandbox Capability determines if you can name sandboxes or if you have one default sandbox:

Description	Sandbox Capability mode
You can name the sandbox and manage multiple sandboxes.	On
Only one default sandbox is available.	Off

The combination of these settings determines how your data changes are stored and processed.

For example, your usergroup may offer direct writeback with named sandboxes. This is the default work design used by TM1. It means you do not have a personal workspace (instead you have direct writeback to the server), but you also have the option of naming a set of changes and manually submitting them. With this setting, when you first open a view, you are in the base and any changes you make are written directly to the base. But, if you decide to save your changes in a named sandbox, you can use the Commit button when you are ready to manually send those changes to update the base.

Consider the case where you usually want to send the data directly to the server. Then you have a set of changes that you want to gather in a group before you update the server. You can use the Create Sandbox options to save the current data changes in a private sandbox called *Best Case*. When you are in the Best Case sandbox, you need to use Commit to send the changes to the base and make the changes available to others. After Best Case is committed, those changes merge with the base so others can see the changes and you are now in the newly updated base. If you are working in a sandbox, it is important to remember that you must manually Commit the sandbox for others to see your changes. Be sure you are ready to make those changes public and that those changes should be merged into the base.

If you move back to the base, you are back to using direct writeback. This setting offers a great deal of flexibility. Users with this setting need to remember when they are updating the base and when the Commit button is needed to make changes available to others.

Or, your administrator may decide that you would like the flexibility to work in a personal workspace writeback mode, but you do not want the complexity of creating named sandboxes. In this case, your Administrator can grant you the personal workspace writeback mode but deny the Sandbox capability.

## Understanding different toolbar options

You can determine how your usergroup is designed to operate based on the options presented on the toolbar. For example, if you have Sandbox granted, you have access to the Create and Delete Sandbox options. When you do not see a sandbox list, you have personal workspace writeback mode.

### Using direct writeback and named sandboxes

By default, IBM TM1 is set to use a direct writeback with named sandboxes. Your Administrator may have set your work options to something different.

You want to	Personal workspace mode	Sandbox mode
Have data changes update the server immediately. Occasionally, you want to save a set of changes and name them before committing them to the server.	Off	On

When you have direct writeback and named sandboxes the toolbar starts out with the Commit and reset Data buttons grayed, the Sandbox button available, and the sandbox list area displays [Base]:



The Sandbox button indicates that you can create and delete sandboxes. The Commit button is grayed but is present because there is nothing to commit yet. If you made a data change and decided to save it in a named sandbox, Commit and Reset Data would become available. Cell coloring would only change when you named a sandbox. Until you name a sandbox, you are operating in the base.

If Job Queuing is turned on, submitting the sandbox to the server is subject to queue processing before the data changes are committed.

## Using a personal workspace and named sandboxes

The personal workspace provides a private work area where users can evaluate data changes before committing the changes to the base. When data is committed, it is merged with the base and becomes available to other users.

Using a personal workspace typically offers a performance improvement over Direct Writeback because users can evaluate their data changes before they commit them, so there is usually less server processing. When Job Queuing is turned on, your personal workspace is subject to processing in the queue before committed changes are merged with the base.

In personal workspace, you begin with the base data. When you make data entry changes, the content that changes, including dependent cells such as consolidations or rule-generated values, change color to blue to remind you that these changes have not yet been merged with the base model. When you commit the personal workspace and processing is complete, the color changes back to black and you are working on the Base again. See [Understanding cell coloring for changed data values](#).

When you have personal workspace granted and the ability to name sandboxes also granted, the starting point for sandbox data is identified in the toolbar as [Default].

You have access to the Commit and Reset Data buttons when you work in personal workspace.

You want to	Personal workspace mode	Sandbox
Always work in a private area and decide when to commit your changes to the server manually. Occasionally, you want to save a set of changes and name them something such as "Best Case" before you commit them to the server.	On	On

When you have personal workspace and named sandboxes, the toolbar includes Commit, Reset Data, Sandbox buttons and the sandbox starting point is called [Default]:



You have the Commit and Reset Data buttons because you are working in a personal workspace. The [Default] sandbox is the way to identify the starting sandbox until you name a sandbox.

## Personal workspace without named sandboxes

If you have access to a personal workspace but do not have the ability to name a sandbox, you do not see the Create and Delete Sandbox buttons and there is no area to list sandboxes since you always work in the same (and single) personal workspace.

You want to	Personal workspace mode	Sandbox
Always work in a private area and decide when to commit your changes to the server manually. You do not want to allow the naming of multiple sandboxes.	On	Off

When you have a personal workspace but do not have the ability to create named sandboxes, the toolbar offers Commit and Reset Data but no sandbox listing area:



Since you always work in the same personal workspace, there are no sandbox names to list but you have access to Commit and Reset Data.

## Direct writeback without sandboxes

This is the classic, direct writeback mode for IBM TM1. In this mode you do not have access to named sandboxes or a personal workspace. You do not have access to the Commit or Reset Data buttons, or have the ability to use Job Queuing. Data changes are not identified by color changes in this option. Data changes in this mode immediately update the server.

To use direct writeback across the entire installation, you can use the `DisableSandboxing=T` setting in the server configuration file. When sandboxing is disabled across the server with this configuration setting, the Capability Assignments are ignored.

You want to	Personal workspace mode	Sandbox
Have your changes take effect immediately in the server. All changes are immediately available to other users.	Off	Off

The toolbar in this case does not have any of the sandbox buttons, Commit, or Reset Data:



You have no access to any kind of sandbox. The only way to take back data changes in this mode is using Undo/Redo.

## Using a personal workspace or sandbox

You can use a sandbox to create your own personal workspace where you can enter and store data value changes separate from base data.

A sandbox is not a copy of the base data, but a separate overlay or layer of your own data values that you enter on top of the base data. This distinction provides a significant performance improvement and is important to understand as you change your data.

- Base data is the data that all users can access. Any edits that are made to base data are written directly back to the database.
- Sandbox data is your own personal work area where you can edit the data values as many times as you want and keep the changed data separate from the base data. Sandboxes and personal workspaces are private to each user and cannot be seen by other users. Your data values are viewable to others only when you commit them back to the base data. A personal workspace is a special, default sandbox that is unnamed and always where you work if that capability is turned on.

Sandboxes are not stored on the client. They consist of a separate and private area of the server. When you work in a sandbox, think of the base model data shining through to the sandbox. When you change data in the sandbox, it is as if the base model data value is temporarily blocked by the value you entered in the sandbox. To make the base model take on the values in the sandbox, you must Commit the sandbox. When the sandbox data values are committed, they are merged with the base so that the changed values then update and become the base values.

Features of sandboxes and personal workspaces include:

- Private data changes.

Sandboxes and personal workspaces let you try out different changes to the data before you make those changes public to other users and before you commit those changes to the base data.

- Cell Coloring.

Changes to cell values in a sandbox or personal workspace are identified by a change in cell content colors. The cells change color to remind you that the change has not yet been merged to the base data. When data is committed and processing is complete, the cell coloring turns to black again.

Cell coloring is also applied to any dependent cells, such as consolidated or rule calculated cells, that your edits affect. For more information, see [“Understanding cell coloring for changed data values” on page 56](#).

- **Queuing.**

Sandbox and personal workspace submissions can be processed using [Job Queuing](#) so jobs that are waiting for resources do not hold up jobs that can be processed right away. The Job Queue also lets you cancel a submission. For more information, see [Canceling a job in the queue](#).

- **Manual Commit.**

When you work in a sandbox or personal workspace, the **Commit** button becomes available so you can decide when to commit changes to the base. When you commit the data, your changes become available to other users.

- **Reset Data.**

In a sandbox or personal workspace, the **Reset Data** button becomes available and lets you return to the status of your sandbox since the last time it was committed.

- **Named sandboxes let you create what-if scenarios.**

Depending on your configuration settings, you can name multiple sandboxes, such as "Best Case" or "Worst Case" and then compare the impact of your edits by switching between them.

- **Virtual sandbox dimension.**

Depending on your configuration settings, you can include sandboxes in a virtual sandbox dimension and compare them in a single view. For example, you can perform side-by-side comparisons of sandbox values and inter-sandbox calculated values.

**Remember:** Your administrator might disable sandboxes for your environment or change the writeback mode for your usergroup.

To work in a sandbox, you must first open a view and then either create a new sandbox or select an existing sandbox. When you work in a sandbox, the selected sandbox applies to all the other views in your current user session.

## **Data values for leaf and consolidated cells in a sandbox**

The data values for leaf and consolidated cells in a sandbox are calculated.

- Leaf cell values in a sandbox are a combination of the values in the base and sandbox cells. The user-entered values in sandbox leaf cells over-ride the values in the base. Any leaf cell that has not been changed in a sandbox still shows the base data.
- Consolidated cells in a sandbox contain values that are the sum of the leaf cells displayed in sandbox.

## **Resetting data values in a sandbox or personal workspace**

Resetting a personal workspace or sandbox or clears all the changed data values that you have entered up to that point and resets all the data values back to the current values in the base data.

### **Procedure**

Depending on which TM1 component you are using:

- In TM1 Web and Server Explorer or Architect, click the Sandbox list and select **Reset Sandbox**.
- In TM1 Perspectives or Microsoft Excel, click the Reset Sandbox button  on the Sandbox toolbar.

### **Results**

All data values in the sandbox are set to the current values in the base data. Any cell coloring is cleared and set to black.

## Understanding cell coloring for changed data values

When you enter a new value in a personal workspace or sandbox, a visual indicator is applied to the cell to remind you that the new value is different from the base values. The color of the data changes from black to either blue or green, or the appearance of the cell changes, depending on which TM1 component you are using. Any dependent cells, such as consolidated or rule calculated cells, also change in appearance if your edits cause them to be recalculated.

The following table summarizes the cell coloring that is applied in the different TM1 user interfaces when you enter new data values in a sandbox or personal workspace.

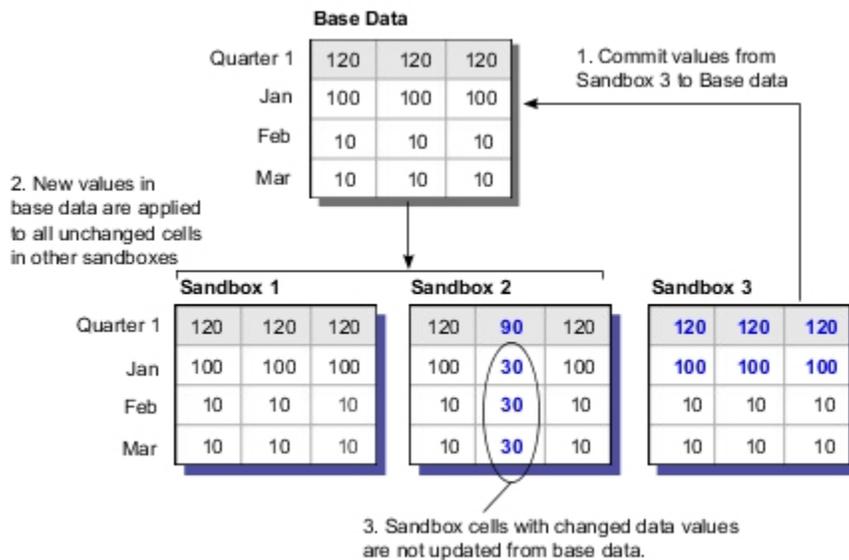
Cell Color	TM1 Component	Direct	Personal workspace or sandbox
Black	TM1 Perspectives / Microsoft Excel Architect Server Explorer	When you input a new value, there is no color change. All values display in black.	Committed personal workspace or Sandbox data.
Blue	TM1 Perspectives / Microsoft Excel Architect Server Explorer	None	Newly input data. Edited cells, dependent or consolidated calls, recalculated cells
Left bottom corner of cell displays in blue	TM1 Perspectives / Microsoft Excel	None	Newly input data. Edited cells, dependent or consolidated calls, recalculated cells
Green	TM1 Web Cube Viewer and websheets	New values	New values

## Committing changed data from a personal workspace or sandbox to base

The Commit command or button merges all of the changed data values in your personal workspace or sandbox to the base data. You cannot use the undo command to undo a commit action.

**Note:** When you have multiple sandboxes and commit one of them to base, the new base values are automatically applied to all the unchanged cells in your other sandboxes. If you entered new data values in any other sandbox, those data values remain and do not show the new values that were committed to the base data.

The following figure shows an example of committing sandbox values to the base data when you are working with multiple sandboxes. In this figure, the new values in Sandbox 3 are committed to base data and then the new base values are applied to all the unchanged cells in the other sandboxes. The figure shows how sandbox cells that contain changed data are not updated.



## Procedure

Depending on which TM1 component you are using:

- In TM1 Web and Server Explorer / Architect, click the Sandbox list and select **Commit Sandbox**.
- In TM1 Perspectives/Microsoft Excel, click the Commit Sandbox button  on the Sandbox toolbar.

TM1 performs the following actions:

- The changed data values in the current sandbox are saved to the base data.
- The cell coloring for any changed data in the current sandbox is cleared and set to black.
- The new base data values are applied to all the unchanged cells in your other sandboxes.

When you have multiple sandboxes, you can use the menu bar pull-down options to create, delete and select the different sandboxes available to you. Some interfaces offer a Delete Sandbox button.

## Job queuing

To maximize processing speed and reduce "traffic jams" when writing back data, Personal Workspace and Sandbox submissions can be processed using a job queue.

To turn on job queuing, your administrator sets the JobQueuing=T parameter in the server configuration file. If this parameter is set to F or not present, Sandbox or Personal Workspace submissions do not use a job queue. In Direct Writeback mode there is no job queuing regardless of this setting. IBM TM1 Contributor does not use the job queue.

The benefits of using submission queuing include:

- Performance improvements.

Use of the queue prevents data that is waiting for resources to hold up other jobs that are ready to process.

- Concurrent work.

The queue allows users to continue working on other jobs while waiting for resources to be freed up on a particular job.

- Transparency of processing.

The queue lets users monitor the activity level in the queue.

- Efficient use of processing resources.

The queue allows users to cancel jobs if necessary.

When the Job Queue is enabled and a Personal Workspace or sandbox is submitted using a Commit or Submit button, the changed data enters the queue as a job and is processed only when the resources needed to complete the calculations specified by the cubes become available. If other sandboxes or Personal Workspace's are submitted while the original sandbox waits for resources, the second submitted sandbox can proceed without waiting for the first one to resolve its resources.

When job queuing is enabled, the job queue button  displays on the toolbar. You can press this button to display the contents of the job queue. You can use the Job Queue and Refresh buttons proactively to see how many jobs are waiting to be submitted or to monitor the progress of a particular submission. Administrators can see all the jobs waiting to be processed in the queue. Users without Admin rights see only their own sandbox submissions.

Queuing progress is based on whether resources are available, not on the amount of data being processed. A submission with a large amount of data that resolves its resources will be processed before a submission with a small amount of data that needs a resource that is in contention.

In many cases sandbox submission will be instantaneous. At times of high concurrent submissions, a user can display the queue and decide to cancel a job. Users can cancel only their own jobs. Administrators can cancel any job in the queue.

When you have submitted a Personal Workspace or sandbox to the job queue:

- In the data, any changed cells remain blue. When the sandbox completes processing, those cells turn black.
- If you have Sandbox turned on, you can create a new sandbox or select an existing one and work as usual, including performing a read, write, or submit. Those submissions will also become subject to the queue. You can even create a new sandbox based on the queued data and work with those values in the new sandbox before the queue processes the transactions.
- You can freely query any data in or out of a sandbox or Personal Workspace, but if you try to update the data, the following message displays so you can indicate your intentions:

```
You are attempting to perform Data Entry while previously committed data changes reside in the queue. Click Yes to remove your submission from the Queue and continue with data entry, or click No to defer your current data entry until the system completes processing of your currently queued job.
```

- To remove your submission from the queue and retain the data changes you just entered, click **Yes**.

When you cancel the job, the data entry is appended to the current sandbox so you can continue working with it and possibly submit it at a later time.

- To wait until the current job completes processing, click **No**.

When you click No, the data entry that is not part of the job is disregarded and the submission continues uninterrupted. Be sure you are willing to lose that data when you click **No** in this situation.

## Viewing the queue

Click the Job Queue button to display the current state of the queue. You can select all jobs or select individual jobs to take action on using the Select check boxes.

There are two tabs in the Job Queue: Active Jobs and Processed Jobs.

Before a job completed processing, it displays in the Active Jobs tab. Everyone can see all active jobs in the queue, not just their own. The information available for each job includes: a selection check box, the relative position in the queue (No.); the user that submitted the job (Client ID); the date and time of the submission (Submission Time); the length of time the job spent in the queue (Duration); and the current status (Pending, for example).

When a job is Pending, you can click the Cancel Job button to cancel the job.

Once a job completes processing, the Processed tab is populated with the job information including the addition of Completion time if the job completed or was canceled. A user can only see their own processed jobs.

Use the Refresh Queue button to update the job submission listing, if necessary.

You can also use Recalc or Refresh and watch for the blue cell coloring to change to black in the sandbox to indicate that the data has been written to the server.

### **Canceling a job in the queue**

Use the Selection check boxes to indicate which job submission to cancel. You can select individual jobs by selecting their individual check box or click the Select All check box to select all jobs currently shown on the Active tab. Click the Cancel job button after you have selected the jobs to remove from the queue.



---

## Chapter 8. TM1 Web and scorecarding

Scorecarding features are integrated into TM1 Web. You can view and interact with scorecarding cubes and diagrams directly within TM1 Web.

Using TM1 Scorecarding, you can:

- Visually monitor organizational strategy and goals
- Monitor your key performance indicators (KPIs) with traffic light status and trend icons
- View and interact with scorecard diagrams and data visualizations

### What is Scorecarding?

Scorecarding is a collection of performance metrics designed to reflect the strategic goals of a business unit or organization. Scorecard information tells you how well the objectives are being met by comparing planned to actual results. Scorecards can also show information for the different organizations in your business. By using visual status indicators such as traffic light and trend icons, scorecards can help you to quickly evaluate business performance.

Scorecarding uses metric dimensions and metric indicator dimensions.

#### Metric

A measure or key performance indicator (KPI) that conveys the performance of an important area of the business. Examples include *Profit*, *Revenue* and *Expenses*.

#### Metric indicator

A measure of performance, status, or trend for a key area (metric) of a business. A metric indicator compares current results to target values. For example, *Score*, *Status*, and *Trend*.

### Working with Scorecard objects in TM1 Web

You can view and interact with the following Scorecarding objects in TM1 Web:

#### Metrics Cubes

A metrics cube is a special type of TM1 cube that provides the basis for scorecard diagrams. This type of cube combines a metrics dimension and a metric indicator dimension with other regular TM1 dimensions, such as time, geography, or products. You can view and analyze scorecard information in a metrics cube using the traffic light status and trend indicator icons that display directly in the cells of the cube.

#### Impact Diagrams

Impact diagrams visualize the positive and negative relationships between the metrics in your metrics cube. This type of diagram shows how one metric impacts another metric.

#### Strategy Map Diagrams

A Strategy Map is an industry standard visualization diagram that combines perspectives, objectives, and metrics with traffic light status and trend indicators icons in one diagram.

#### Custom Diagrams

A Custom scorecard diagram is a strategy map that combines metrics with a custom image. The metrics are displayed with dimensional context on top of the image as data points.

### Scorecarding Modeling

Scorecarding objects are created in IBM TM1 Performance Modeler. For more information contact your administrator or see the *TM1 Performance Modeler* documentation.

## Scorecarding Samples

The TM1 installation provides a scorecarding database sample called G0\_scorecards. For more information about using this sample, contact your administrator or see the *Planning Analytics Installation and Configuration* documentation.

## Scorecarding objects in TM1 Web

You can display and interact with Metrics Cubes, Impact Diagrams, and Strategy Map Diagrams in TM1 Web.

### Metrics cubes in TM1 Web

IBM TM1 Web displays metrics cubes so you can view and analyze scorecard information.

A metrics cube is a special type of TM1 cube that provides the basis for scorecard diagrams. This type of cube combines a metrics dimension and a metric indicator dimension with other regular TM1 dimensions, such as time, geography, or products.

The main difference between a metrics cube and a standard cube, is that the metrics cube displays traffic light status and trend indicator icons directly in the cells of the cube. These metric indicator icons show the status and trend of each metric in the cube.

A standard scorecard layout for a metrics cube is:

- **Row title dimension:** Metrics dimension
- **Column title dimension:** Metric Indicator dimension
- **Required context dimension:** Time
- **Optional context dimensions:** Geography, products, or other data context dimensions.

	Status	Trend	Actual	Target	Tolerance	Variance
Gross profit	●	↓	\$4,777,074	\$4,078,288	5%	\$698,786
Gross margin	●	↓	90.057%	84.268%	5%	5.789%
Calculated Bonus	■	↓	364.16	80158	5%	-437.43
Bonus	■	↓	0.95%	1.25%	10%	(0.30%)
Revenue	●	↓	\$5,282,604	\$4,805,824	5%	\$476,780
Product cost	●	↑	\$150,640	\$161,055	5%	(\$10,416)
Quantity sold	●	▬	481,930	314,150	5%	167,780
Employee expense	●	↑	\$3,554	\$5,207	5%	(\$1,653)
Salary	●	↑	\$233,472	\$329,972	5%	(\$96,500)
Training cost	●	↓	\$117,500	\$230,500	5%	(\$113,000)

1. gometric dimension
2. Metric indicators dimension
3. Context dimension
4. Time dimension
5. Traffic light status indicator icons
6. Trend indicator icons

## Metric indicators

The metric indicators in a metrics cube measure the performance, status, and trends in key areas of a business by comparing current results to target values. For example, the *Actual*, *Target*, and *Tolerance* indicators for a metric are typically used to calculate the related *Score*, *Status*, and *Trend* indicators.

Metric indicators can be shown as numeric values or visually as traffic light and trend icons. The Metric Indicator dimension is typically shown in the column dimension title of a scorecard cube view.

### Traffic light status indicator

A traffic light or status indicator is an icon that shows the status of a Metric indicator. The status is indicated by the color and the shape of the icon as described in the following table.

Traffic light icon	Description
	A green circle icon indicates a satisfactory status for the associated Metric indicator.
	A yellow diamond icon indicates caution about the status for the associated Metric indicator.
	A red square icon indicates a warning about the status for the associated Metric indicator.
	This image represents an incomplete status for when there is no data for the Actual or Target Metric indicators. A score or status cannot be calculated when one of these values is missing.

### Trend indicator

A trend indicator shows how the value of one column compares to the value of another column. The trend indicator shows if the value is greater than, unchanged, or less than the other value.

Trend icon	Description
	A green upward facing triangle icon indicates that the trend value is greater than the previous period. For example, the value is greater than the previous month or quarter.
	A gray dash icon indicates that the trend value is unchanged.
	A red downward facing triangle indicates that the trend value is less than the previous period. For example, the value is less than the previous month or quarter.
Blank cell	Indicates that the trend is incomplete for that period. A trend cannot be displayed when there is an incomplete status. For example, a trend cannot be displayed for the first time period, such as Q1 (quarter one) since previous data does not exist, even if the metric has a value for Actual, Target, Score, and Status.

## Impact diagrams in TM1 Web

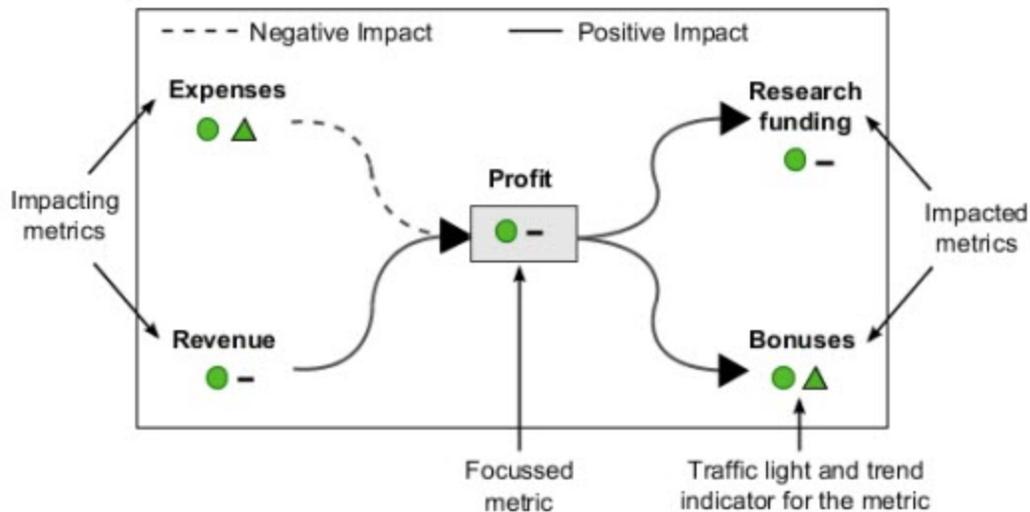
Impact diagrams visualize the positive and negative relationships between the metrics in your metrics cube. This type of diagram shows how one metric impacts another metric.

**Note:** By default, only one impact diagram can exist for a metrics cube.

Impact diagrams organize your metrics into three categories:

- *Impacting Metrics*
- *Focused Metrics*
- *Impacted Metrics*

For example, an impact diagram might show how *Revenue* and *Expenses* impact *Profit*, which then impacts *Bonuses* and *Research Funding*.



The lines in the diagram show the impact relationships between the metrics in the diagram. These lines show if a metric has a positive or negative impact in relation to the focused metric.

- **Solid line** - represents a positive impact from one metric to another metric.
- **Dashed line** - represents a negative impact from one metric to another metric.

Impact diagrams also display traffic light and trend indicator icons to show the status and the trend of each metric in the diagram.

## Strategy maps in TM1 Web

A Strategy Map is an industry standard visualization diagram that tracks business performance by *perspectives, objectives and metrics*.

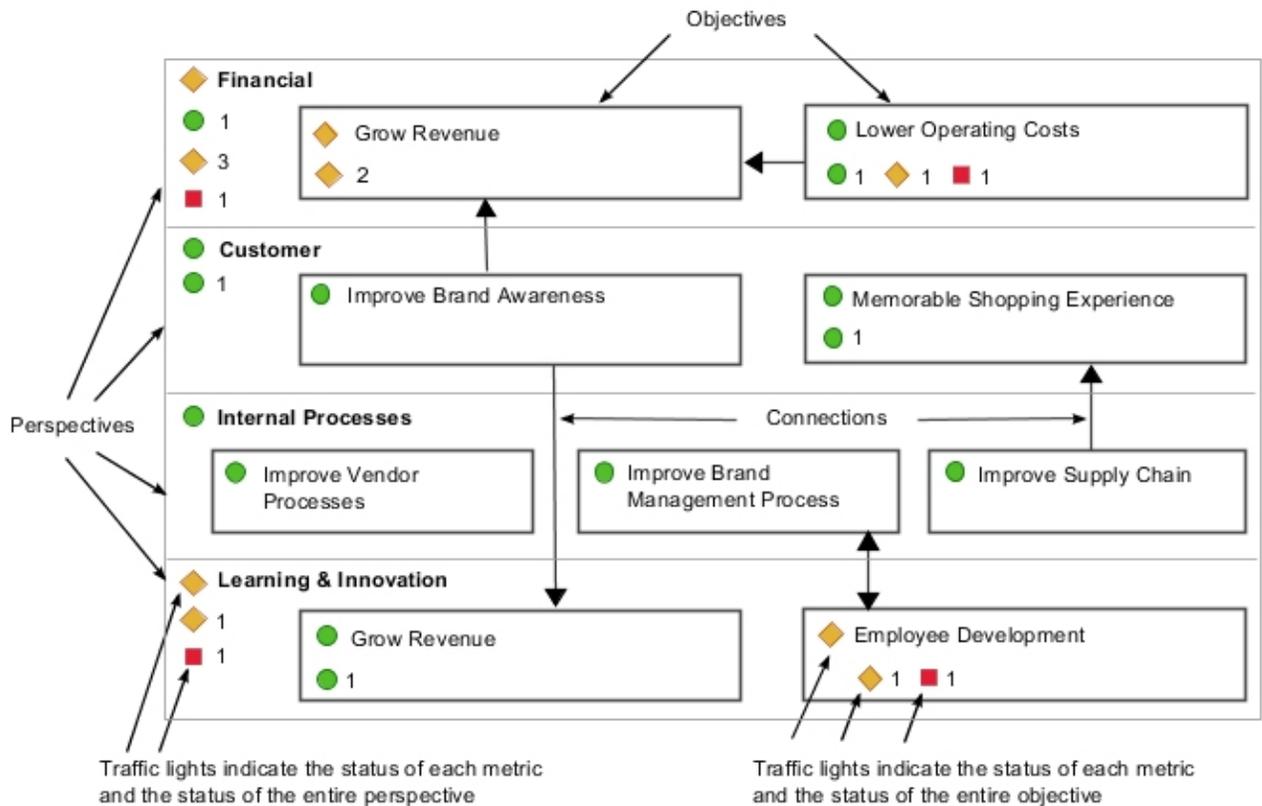
A Strategy Map organizes perspectives, objectives, and metrics into the following hierarchy:

- A Strategy Map can have multiple perspectives.
- Each perspective can have multiple objectives.
- Each objective can have multiple metrics.

### Perspectives

The standard perspectives for a Strategy Map include:

- Financial
- Customer
- Internal Processes
- Learning and Innovation



### Status and trend indicators

A Strategy Map combines perspectives, objectives, and metrics with traffic light status and trend indicators icons into one diagram. When you hover your mouse over an objective, a detailed list of the status and trend for the related metric indicators is displayed. Hovering your mouse over the indicator icons for a perspective shows the name of the diagram and perspective.

### Connections

A Strategy Map can also display directional arrows called connections to show a visual relationship or flow between the objectives in the diagram.

## Custom diagrams in TM1 Web

You can display a Custom scorecard diagram in TM1 Web using the chart feature. A Custom scorecard diagram displays your metrics with dimensional context as data points on top of a custom image.

Some examples of a Custom diagram are identified in the following list:

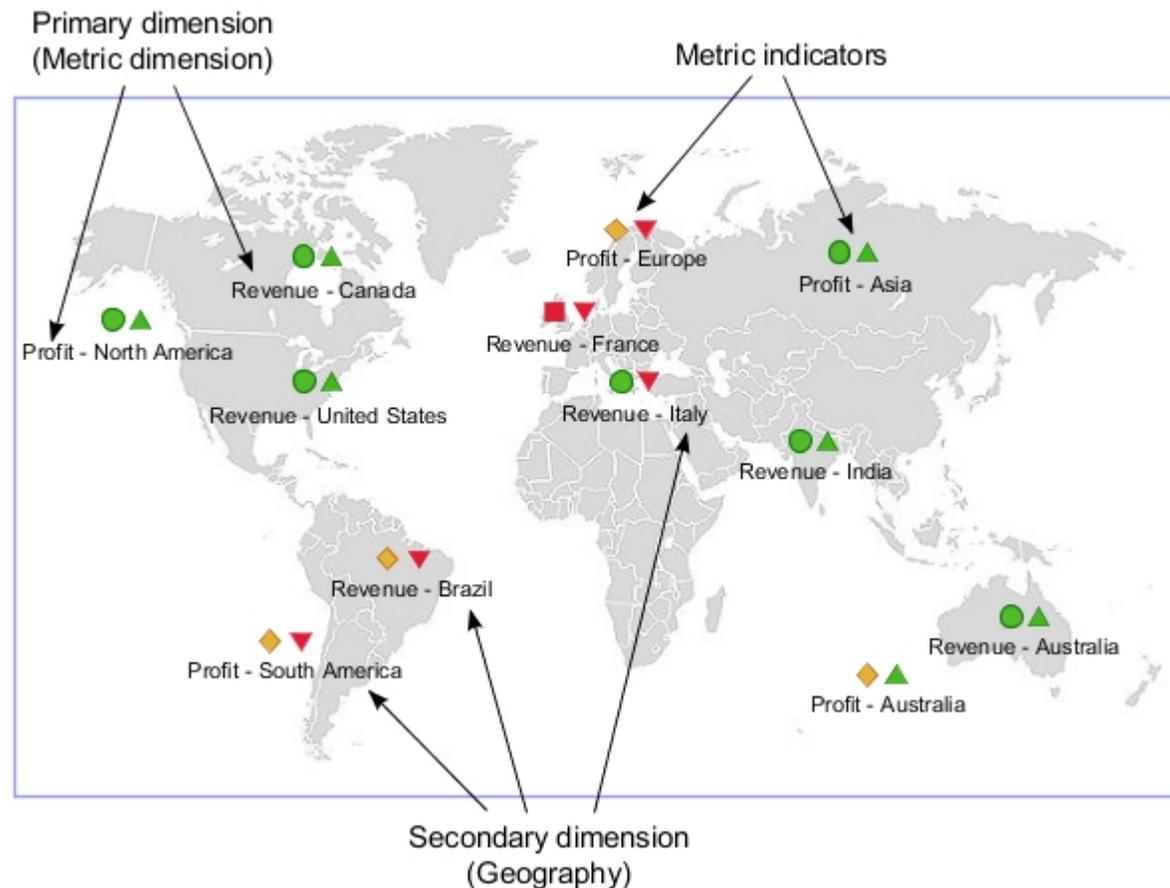
### Geographical map

Shows metrics for different focuses of your organization on locations or regions, such as inventory or cost metrics in North America or Europe.

### Process diagram

Shows metrics in the context of a process flow.

A Custom diagram displays the metric and context dimension names with traffic light and trend indicator icons as an overlay on top of an image. When you hover the mouse cursor over a metric, a pop up window shows more data for that point.



## Viewing metrics cubes in TM1 Web

You can view a scorecarding metrics cube in IBM TM1 Web just like any other TM1 cube or view. Metrics cubes are listed in the TM1 Web Navigation pane along with all the other TM1 cubes and views in the server you have logged into.

### Procedure

1. In the TM1 Web Navigation pane, locate the metrics cube that you want to open and do one of the following:
  - Double-click the cube to open the default view.
  - Expand the node for that cube and click a specific view.

The metrics cube opens and displays traffic light and trend indicator icons in the cell values to show the status and trend of each metric in the cube.

2. Use the **View Chart** and **Chart Properties > Metric Diagram** icons in the toolbar to view the related scorecarding diagrams for the metrics cube.

For more information, see [“Viewing impact diagrams in TM1 Web”](#) on page 67 and [“Viewing strategy maps in TM1 Web”](#) on page 67.

## Viewing impact diagrams in TM1 Web

---

You can view scorecarding impact diagrams in IBM TM1 Web.

### Before you begin

The TM1 server that you are using must contain at least one scorecard metrics cube in order to display this type of diagram.

### Procedure

1. Open a metrics cube view.

For more information, see [“Viewing metrics cubes in TM1 Web” on page 66](#).

2. Change the TM1 Web layout to display a chart.

In the toolbar, click either the **View Chart** or **View Chart and Grid** icon.

3. Click **Chart Properties > Metric Diagram > Impact Diagram**.

The impact diagram for the current metrics cube view displays.

**Note:** By default, a metrics cube can have only one impact diagram so there is only one to select.

4. In the diagram, hover the mouse cursor over a metric to see information about the metric indicators for that metric.
5. Click the plus (+) and minus (-) icons next to a metric to expand and collapse sections of the diagram.
6. Use the TM1 Web Subset Editor to change the focussed metric in the diagram to a different metric.
  - a) In the cube view, click **Open Subset Editor** ▼ next to the metrics title dimension. The Subset Editor opens.
  - b) Drag the metric that you want to use as the focussed metric into the **Subset** pane.
  - c) In the **Subset** pane, click the metric that you want to use.
  - d) Click **OK**.

The impact diagram updates to show the selected metric as the focussed metric.

## Viewing strategy maps in TM1 Web

---

You can view scorecarding Strategy Map diagrams in IBM TM1 Web.

### Before you begin

The TM1 server that you are using must contain at least one scorecard metrics cube which must also contain one or more Strategy Map diagrams for that cube.

### Procedure

1. Open a metrics cube view.

For more information, see [“Viewing metrics cubes in TM1 Web” on page 66](#).

2. Change TM1 Web layout to display a chart.

In the toolbar, click either the **View Chart** or **View Chart and Grid** icon.

3. Click the **Chart Properties > Metric Diagram** and select one of the Strategy Map diagrams.

**Note:** A metrics cube can contain one or more Strategy Map diagrams.

4. In the toolbar, click the **View Chart** icon to display the diagram in full-size mode.
5. Hover the mouse over the perspectives and objectives in the diagram to see more details.

## Viewing custom diagrams in TM1 Web

---

You can view Custom Scorecarding diagrams in IBM TM1 Web using the chart feature.

### Before you begin

The TM1 server that you are using must contain at least one scorecard metrics cube which must also contain one or more Custom diagrams for that cube.

### Procedure

1. Open a metrics cube view.

For more information, see [“Viewing metrics cubes in TM1 Web”](#) on page 66.

2. Change TM1 Web layout to display a chart.

In the toolbar, click either the **View Chart** or **View Chart and Grid** icon.

3. Click the **Chart Properties > Metric Diagram** and select one of the available Custom diagrams.

**Note:** A metrics cube can contain one or more Custom diagrams.

4. In the toolbar, click the **View Chart** icon to display the diagram in full-size mode.
5. Hover the mouse cursor over a metric data point in the diagram to see more details for that metric.

---

## Chapter 9. Administering IBM TM1 Web

You can configure IBM TM1 to work over the Web.

### IBM TM1 Web overview

---

IBM TM1 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 TM1 Web

---

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

#### Procedure

1. On the 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.

If no path is specified, the `relational.key` file appears in the following location:

`tm1web\WEB-INF\cert\key\relational.key`

**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.

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 TM1 Web configuration parameters

---

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

**Remember:** In IBM Planning Analytics Local version 2.0.6 and later, you install a new version of the `tm1web_config.xml` file that is called `tm1web_config.xml.new` and your existing `tm1web_config.xml` file is preserved. To take advantage of fixes that are applied to the 2.0.6 release, you must use the `tm1web_config.xml.new` file and you must reapply any changes that you made to your previous configuration settings.

The parameters in this file control the following IBM TM1 Web features.

- View node
- Cube Viewer page size
- Number of sheets to export from a Cube Viewer

- IBM TM1 Web startup and appearance settings
- Session timeouts

## TM1 Web configuration parameters

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

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

```
<TM1 install location>\webapps\tm1web\WEB-INF\configuration\
```

The following parameters are available.

### 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 TM1 Web login page using AdminHostName and TM1ServerName parameters” on page 77.](#)

### 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 worksheet 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 85.](#)

#### **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.

#### **CubeViewerRowPageSize**

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

See [“Changing the Cube Viewer page size” on page 85.](#)

#### **CubeviewerStringWrap**

Settings for string cell wrapping in the Cubeviewer.

See [“Wrapping string values in cube views” on page 85.](#)

#### **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 worksheet 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 84.](#)

### HideTabBar

If set to true, multiple tabs are not displayed.

See [“HideTabBar parameter” on page 83.](#)

### HideWebsheetToolBar

If set to true, all websheet toolbars are not displayed.

See [“HideWebsheetToolBar parameter” on page 84.](#)

### 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 80.](#)

### HttpSessionTimeout

This parameter defines the session timeout (in minutes) of the HTTP session for TM1 Web. If the parameter is missing, the value is less than 1, blank, or not a numerical value, the default session timeout that is defined for TM1 Web in the web.xml file is used.

See [“Setting the TM1 Web session timeout” on page 86.](#)

**Note:** As of IBM Planning Analytics Local version 2.0.6, you should not change the session-timeout value in the web.xml file.

### 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.

See [TM1 Web API session login.](#)

Use this format:

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

For example:

```
<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 85.

### **MixedCellPaste**

If the MixedCellPaste parameter is set to true, when you copy values to a mixed range of leaves and consolidated values in a worksheet, the pasted values will match exactly. The default value is False.

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

### **NavTreeCollapsedOnStart**

Determines whether the navigation panel will be collapsed or expanded after a user logs in.

See [“NavTreeCollapsedOnStart parameter”](#) on page 83.

### **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 84.

### **NavTreeHidden**

Determines whether the navigation panel will be displayed after a user logs in.

See [“NavTreeHidden parameter”](#) on page 82.

### **RecalcOnActivate**

If RecalcOnActivate is set to true, a recalculate is performed each time a worksheet 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 84 in *Configuring IBM 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 TM1 Web login page using AdminHostName and TM1ServerName parameters" on page 77.

### **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:

```
<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.

## WorksheetBackgroundRecalculationMode

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

WebSheetService.scrollWebSheet calls can take several seconds because the data is not readily available. Use the WorksheetBackgroundRecalculationMode 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.

See [Using IBM 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 TM1 Web configuration file

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

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

As of TM1 Web version 10.2, the new `tm1web_config.xml` file replaces the `web.config` file from previous 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 TM1 Web to see the result of your edits.

## Configuring the TM1 Web login page using AdminHostName and TM1ServerName parameters

The AdminHostName and TM1ServerName parameters control whether the IBM 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:

```
<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 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:

```
<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 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 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 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.

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

## Configuring a custom homepage for TM1 Web

You can configure a custom homepage for IBM TM1 Web to display a websheet, cube view, or a URL after users have successfully logged into IBM 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 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 TM1 Web and cannot be closed by users. When configured, a Home link is displayed in the header area of IBM TM1 Web that allows users to easily return to the homepage.

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

### Different homepage for different IBM 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 TM1 Web.

### Global homepage for all IBM TM1 Web users

Use the HomePageObject parameter in the tm1web\_config.xml file to configure a homepage that applies globally to all IBM 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 TM1 Web clients (users).

For example, you can assign one homepage for 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 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 more information, see [“Configuring TM1 Web startup and appearance settings”](#) on page 82.

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 TM1 Web.
  - **Open pane on Login** - Sets the Navigation pane to display in the expanded mode when the selected client logs in to TM1 Web.
  - **Close pane on Login** - Sets the Navigation pane to display in its minimized mode when the selected client logs in to 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 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.

The available options include:

- **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.

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.

- **All Clients** - Applies the same homepage setting to all TM1 clients.
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 TM1 Web. The selected TM1 Web clients will see the assigned homepage the next time they successfully log in to 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 TM1 Web users.

**Note:** You can override the global `HomePageObject` parameter by using the Client Settings dialog to assign different homepages for individual TM1 users. For more information, see [“Configuring different homepages for individual users”](#) on page 78

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

- Cubeviewer
- Websheet
- URL

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

### Using the `HomePageObject` parameter

How to use the `HomePageObject` parameter.

The **`HomePageObject`** parameter uses the following format:

```
<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 TM1 Web homepage to a Cube View

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

```
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:

```
&lt;add key="HomePageObject" value="SalesCube$$Price$$Public;Type=cubeviewer;  
Description=MyStartCube;AllowOverwrite=true"  
>
```

### Setting a global TM1 Web homepage to a worksheet

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

### Opening a worksheet that references an Excel file outside of TM1

You can open a worksheet that references an Excel file.

### Procedure

Use the format:

```
value="WorksheetPath"
```

where **WorksheetPath** 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 worksheet:

```
value=//MySystem/Samples/classic_slice.xls
```

### Results

The complete HomePageObject parameter looks like this:

```
<add key="HomePageObject" value="//MySystem/Samples/classic_slice.xls;Type=worksheet;
```

```
Description=MyWorksheet;AllowOverwrite=true"/>
```

### Opening a worksheet object that was uploaded to the TM1 server

You can open a worksheet object that was uploaded.

### Procedure

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

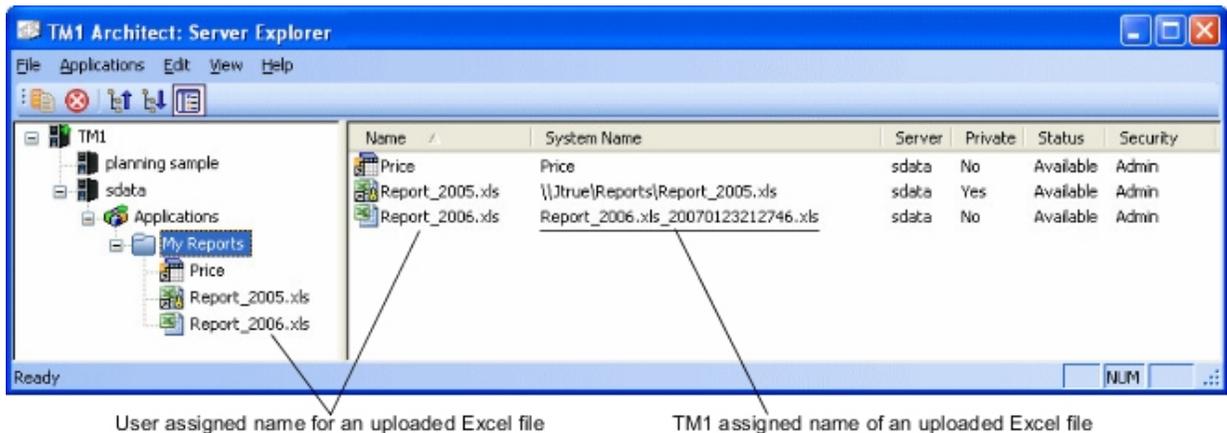


Figure 1: Example of an assigned name for an uploaded Excel file in Server Explorer

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 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:

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

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

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

## **Configuring 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 TM1 Web.

These parameters are located in the `tm1web_config.xml` file and apply globally to all users of TM1 Web.

**Note:** For more information on using the HomePageObject parameter to set a custom homepage, see [“Configuring a custom homepage for TM1 Web” on page 78](#).

### **NavTreeHidden parameter**

The NavTreeHidden parameter determines if the Navigation pane displays when users log in to IBM 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 TM1 Web.
- If set to true, the Navigation pane will not be displayed when user's log in to TM1 Web.

AllowOverwrite can be set to true or false as follows:

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

### NavTreeCollapsedOnStart parameter

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

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

```
<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 TM1 Web.
- If value is set to true, the Navigation pane will be collapsed when user's log in to TM1 Web.

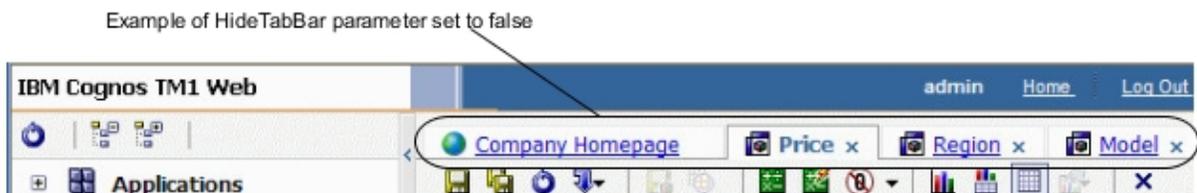
`AllowOverwrite` can be set to true or false as follows:

- 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 TM1 Web can display multiple tabs when a user opens multiple TM1 Web objects, or if only one view is displayed.

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



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

```
<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 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:

```
<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 TM1 Web.
- If value is set to true, the websheet toolbar will not display in TM1 Web.

The AllowOverwrite option is not currently used for this parameter.

### HideCubeviewerToolBar parameter

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

The HideCubeviewerToolBar parameter uses the following format in the tm1web\_config.xml file:

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

where value can be either true or false.

- If value is set to false, the websheet toolbar will display in TM1 Web.
- If value is set to true, the websheet toolbar will not display in 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 TM1 Web virtual directory.
2. Locate the NavTreeDisplayServerView, which controls the display of the **Server View** node. The default value, Y, displays the **Views** node in the Navigation pane.

```
<!--NavTreeDisplayServerView: Y/N - Wether to display  
"Server View" node in navigation tree -->
```

```
<add key="NavTreeDisplayServerView" value="Y" />
```

3. To hide the Views node, change the NavTreeDisplayServerView value to N.
4. Save tm1web\_config.xml.
5. Log in to 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:

```
<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 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 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 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 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 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).

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

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

## Setting the TM1 Web session timeout

The default TM1 Web session timeout is 20 minutes. When TM1 websheets are deployed to IBM Planning Analytics Workspace, you might encounter TM1 Web session timeouts. You can modify this setting in your environment.

### About this task

When TM1 websheets are deployed to Planning Analytics Workspace, the recommended session timeout is 60 minutes.

**Note:** As of IBM Planning Analytics Local version 2.0.6, you must **not** change the `session-timeout` value in the `web.xml` file.

In IBM Planning Analytics Local version 2.0.6, there is a parameter in the `tm1web_config.xml` file called `HttpSessionTimeout`. You can use this parameter to customize the session timeout (in minutes) of the HTTP session for TM1 Web.

If the `HttpSessionTimeout` parameter is not specified (missing or blank), the value is less than 1 or not a numerical value, the default `session-timeout` that is defined in the `web.xml` file is used.

If you are using IBM Planning Analytics Local version 2.0.6 or later, to customize the session timeout for TM1 Web, set the `HttpSessionTimeout` parameter in `tm1web_config.xml`. See step [“1” on page 86](#).

If you are using IBM Planning Analytics Local version 2.0.5 or earlier, to change the default session timeout, set the `<session-timeout>` parameter in `web.xml`. See step [“2” on page 86](#).

### Procedure

1. To customize the session timeout, follow these steps.

a) Open the `tm1web_config.xml` in a text editor.

The `tm1web_config.xml` file is located in your `<pa_installation_directory>\webapps\tm1web\WEB-INF\configuration` directory.

For example, `C:\Program Files\IBM\cognos\tm1_64\webapps\tm1web\WEB-INF\configuration`.

b) Change the `HttpSessionTimeout` to 60 or a value that is required by your environment.

```
<add key="HttpSessionTimeout" value="60" />
```

c) Save and close the `tm1web_config.xml`.

2. If you are using IBM Planning Analytics Local version 2.0.5 or earlier, to change the default session timeout, follow these steps.

a) Open `web.xml` in a text editor.

The `web.xml` file is located in your `<pa_installation_directory>\webapps\tm1web\WEB-INF` directory.

For example, `C:\Program Files\IBM\cognos\tm1_64\webapps\tm1web\WEB-INF`.

b) Change the <session-timeout> value to 60 or a value that is required by your environment.

```
<session-config>  
  <session-timeout>20</session-timeout>  
</session-config>
```

c) Save and close web.xml.

3. Restart the IBM TM1 Application Server service.

## Using TM1 Web logging

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

### IBM TM1 Web log file

The logging process for IBM 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 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 TM1 Web logging

The logging process for IBM 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.

Parameter	Description
DEBUG	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.
INFO	Informational messages that highlight the progress of the application and report normal transitions within the application. When logging is configured to this level, INFO and ERROR messages are logged.
ERROR	An error condition of which you should be aware. Action should be taken to fix or report the issue to TM1 customer support. When logging is configured to this level, only ERROR messages are logged.

## Configuring and enabling IBM TM1 Web logging

You can change the logging message level for IBM 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 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 Customer Support.

The following is a sample of the logging properties file.

```
# System logging settings
log4j.rootLogger=ERROR, TextFile
log4j.logger.com.ibm.cognos=ERROR
log4j.logger.com.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=org.apache.log4j.PatternLayout
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 TM1 Web log file

The IBM TM1 Web installation configures IBM 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 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.

## What if I use Microsoft Excel 2007 or earlier .xls worksheets?

---

IBM 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 TM1 conversion tool to convert the files. If your original file contained macros, the 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:

```
<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.

---

## Appendix A. TM1 Web API

In addition to using IBM TM1 Web as a stand-alone application, you can also use it in your own custom web applications. Web programmers and TM1 application developers can use the TM1 Web application programming interface (API) to incorporate TM1 Web objects into custom web pages, applications, and dashboards.

The TM1 Web API includes two separate sets of APIs. These APIs also share a common login approach that uses session tokens or TM1 session IDs.

Depending on your specific development requirements, you can choose between the two different APIs and use the same login approach with either one.

### TM1 Web API session login

The TM1 Web APIs share a common login approach that uses session tokens to uniquely identify and separate your TM1 Web sessions or TM1 session IDs to uniquely identify your TM1 Server. You can use this login approach with both APIs.

For more information, see [“TM1 Web API session login” on page 91](#).

### TM1 Web URL API

The URL API provides access to Websheet and CubeViewer objects by using a special set of URLs and parameters. Simple examples can be done right in the address bar of a web browser. To create a solution with the URL API, you need knowledge of HTML and an optional knowledge of JavaScript.

See [“TM1 Web URL API” on page 97](#).

### TM1 Web JavaScript Library

The JavaScript Library enables programmatic access to TM1 Web Websheet and CubeViewer objects in a combined HTML, JavaScript, and Dojo web page development environment. To use the JavaScript Library you need knowledge of HTML, JavaScript, Dojo, and the HTML Document Object Model (DOM).

See [“TM1 Web JavaScript library” on page 118](#).

---

## TM1 Web API session login

Use the session token login approach to uniquely identify your TM1 Web session. This login approach is recommended for the URL API. Use the TM1 session ID login approach to uniquely identify a TM1 server session, which might have multiple TM1 Web sessions. Use the session and login modules for easier session management with the JavaScript library.

### Session token login

The session token login returns a unique session token that represents a login session for a specific user, Admin host, and TM1 server combination.

**Important:** Each TM1 Web session is associated with an HTTP session. The TM1 Web session token can be used only under the HTTP session in which it was created. You cannot save a TM1 Web session token, open a browser on another device, and get access to the TM1 Web session that corresponds to that session token because the HTTP session is different.

You can use the JavaScript XMLHttpRequest API to send an HTTP login request to the TM1 Web server. The session token is then returned in a JavaScript Object Notation (JSON) format from the request. After you receive the session token, you can then use it when you open TM1 Web objects.

If a timeout occurs with your HTTP session from inactivity, the TM1 Web session and related token are no longer valid.

## TM1 Session ID login

Users can also log in by specifying a TM1 server session with a `TM1SessionId`. The TM1 server session that is used by a TM1 Web session will never change and must be generated or specified on creation. Multiple TM1 Web sessions can use the same TM1 server session.

## Session and Login modules

In the JavaScript library, you can use the `session` and `LoginDialog` APIs to manage sessions and login dialog boxes.

For more information, see [“Session and LoginDialog modules” on page 94](#).

## Session token login

The overall process for logging in with a session token includes the following steps.

1. If you are using the URL API, set the `LegacyUrlApiSessionDiscoveryEnabled` configuration parameter in the `tm1web_config.xml` file.  
**Note:** This configuration parameter is not needed if you are using the JavaScript library.
2. Assemble a set of parameters for the login request that are based on the type of authentication you are using with TM1.
3. Post the login request to the TM1 Web server by using the JavaScript `XMLHttpRequest` API or other similar approach.
4. Process the JSON response to get the returned session token.
5. Use the session token when you open `Websheet` and `CubeViewer` objects.

### Configuration parameter for session token login

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`.

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

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

### Login request parameters

Use the session token approach by sending a set of parameters in the request for the type of authentication that you are using with TM1.

For TM1 standard authentication and integrated login, use the following parameter format:

- `param0=TM1_Admin_host`
- `param1=TM1_server_name`
- `param2=username`
- `param3=password`

For example:

```
param0=localhost&param1=SData&param2=admin&param3=apple
```

If you are using IBM Cognos Analytics security for authentication, use the following format to include a value for the `camPassport`:

- `param0=TM1_Admin_host`
- `param1=TM1_Server_name`
- `param2=camPassport`

## JSON reply for session token login

The results of the login request are returned in a JSON formatted string.

If the login request is successful, the reply is returned in the following format.

```
{
  "reply": {
    "adminHost": adminHost,
    "sessionToken": sessionToken,
    "tm1Server": tm1Server,
    "username": username
  }
}
```

For example:

```
{
  "reply": {
    "adminHost": "localhost",
    "sessionToken": "06974cbd-ff2d-408b-8181-87bdd3f9048",
    "tm1Server": "Planning Sample",
    "username": "admin"
  }
}
```

If the login request is not successful, the following reply is returned.

```
{ "reply": null }
```

## Example

The following example uses the JavaScript XMLHttpRequest API to post a login request to the TM1 Web server and retrieve the assigned session token.

```
<script type="text/javascript">

function login() {
  var xhr = new XMLHttpRequest();
  xhr.open("POST", "http://localhost:9510/tm1web/api/TM1Service/login", true);
  xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhr.onload = function() {
    var response = JSON.parse(xhr.responseText).reply;

    if(response != null) {
      var sessionToken = response.sessionToken;
      console.debug("Session token: " + sessionToken);
    }
    else {
      console.error("Login failed.");
    }
  }

  var params = "param0=localhost&param1=Planning+Sample&param2=admin&param3=apple";

  xhr.send(params);
};

</script>
```

## LegacyUrlApiSessionDiscoveryEnabled configuration parameter

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” on page 91](#).

## Format

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

For example:

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

## Values

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.

## TM1 session ID login

Users can log in by specifying a TM1 server session with an admin host, TM1 server name, and `TM1SessionId`. The `TM1SessionId` corresponds to a user session on a TM1 server. To retrieve data from a TM1 server, a valid user session is required. Every TM1 Web session requires a TM1 server session. The overall process for logging in with a TM1 session ID is similar to the process for logging in with a session token except the `TM1SessionID` parameter replaces the `sessionToken` parameter:

```
TM1SessionId=valid TM1 session ID
```

This login method creates a new TM1 Web session and reuses the TM1 server session that corresponds to the `TM1SessionId`. If a TM1 server session is shared between TM1 Web sessions, invalidating the TM1 server session results in the TM1 Web sessions also being invalidated.

## Example

In the following example, a `TM1SessionId` parameter is included in the URL to support this type of login authentication.

```
http://localhost:9510/tm1web/UrlApi.jsp#Action=Open&Type=WebSheet&Workbook=Applications/  
Planning Sample/Bottom Up Input/Budget Input&AdminHost=localhost&TM1Server=Planning  
Sample&TM1SessionId=<valid TM1 session ID>
```

## Session and LoginDialog modules

You can use the session and `LoginDialog` APIs to easily manage user sessions and login dialogs with the JavaScript library.

### Session

You can use `tm1web/api/session/session` to retrieve information that is associated with the TM1 Web session. You can log in to, log out of, or retrieve information for a TM1 Web session.

### Methods

#### login(params)

Performs a login to TM1 Web.

Parameters: `params` The login information object that uses one of the following object formats:

```
{
  adminHost: "localhost",
  tm1Server: "Planning Sample",
  username: "admin",
  password: "apple"
}
```

Or

```
{
  adminHost: "localhost",
  tm1Server: "Planning Sample",
  camPassport: "8sdf83uijsjdfsd903sd"
}
```

Or

```
{
  adminHost: "localhost",
  tm1Server: "Planning Sample",
  tm1SessionId: "D31JLw50uvh2jtbAcIYyVA"
}
```

Returns `dojo/promise/Promise` as a promise that is resolved when the login action completes. If login fails, the promise is rejected, otherwise it is resolved. The promise is passed an object of the following format if login is successful.

```
{
  sessionToken: "7118fad5-bbeb-4b3e-8bea-4b4a45ca2735",
  tm1SessionId: "D31JLw50uvh2jtbAcIYyVA",
  adminHost: "localhost",
  tm1Server: "Planning Sample",
  username: "Admin"
}
```

### **getInfo(sessionToken)**

Retrieves the information that is associated with the TM1 Web session corresponding to the specified session token.

Parameters: `sessionToken` A session token corresponding to the TM1 Web session to retrieve information from.

Returns `dojo/promise/Promise` as a promise that is resolved when the action completes. If retrieval fails, the promise is rejected, otherwise it is resolved. The promise is passed an object of the following format if retrieval was successful.

```
{
  sessionToken: "7118fad5-bbeb-4b3e-8bea-4b4a45ca2735",
  tm1SessionId: "D31JLw50uvh2jtbAcIYyVA",
  adminHost: "localhost",
  tm1Server: "Planning Sample",
  username: "Admin"
}
```

### **logout(sessionToken)**

Performs a logout and invalidates the TM1 Web session corresponding to the specified session token.

Parameters: `sessionToken` A session token corresponding to the TM1 Web session to invalidate.

Returns `dojo/promise/Promise` as a promise that is resolved when the action completes. If retrieval fails, the promise is rejected, otherwise it is resolved. The action completes successfully even if the session does not exist or has already been invalidated.

For more information, see [Dojo documentation for dijit.\\_WidgetBase](https://dojotoolkit.org/reference-guide/1.10/dijit/_WidgetBase.html) (https://dojotoolkit.org/reference-guide/1.10/dijit/\_WidgetBase.html).

## Examples

```
// login
require([
  "tm1web/api/session/session"
], function(session) {
  session.login({
    adminHost: "localhost",
    tm1Server: "Planning Sample",
    username: "admin",
    password: "apple"
  }).then(function(sessionInfo) {
    // Create Workbook or CubeViewer using sessionInfo.sessionToken
  }, function() {
    // Handle login failure appropriately
  });
});
```

```
// getInfo
require([
  "tm1web/api/session/session"
], function(session) {
  session.getInfo("sessionToken").then(function(sessionInfo) {
    // Continue using obtained sessionInfo
  });
});
```

```
// logout
require([
  "tm1web/api/session/session"
], function(session) {
  session.logout("sessionToken").then(function() {
    // Logout has successfully completed
  });
});
```

## LoginDialog

You can use `tm1web/api/session/LoginDialog` to display or destroy a login dialog box.

## Example

```
var dialog = new LoginDialog({
  onLogin: function(sessionInfo) {
    console.log(sessionInfo);
  },
  tm1Server: "Planning Sample",
  adminHost: "localhost"
});

dialog.show();
```

## Construction

The `LoginDialog` module accepts several parameters for construction.

### onLogin

Type: Function

Callback when login succeeds. An object that contains session information is passed as a parameter to the function on execution.

An example of this object is:

```
{
  tm1SessionId : "JcFxnISEzsJZV1QQhYDLQ",
  sessionToken : "baa4ff9a-ddfb-41d1-9c71-f0add92325fd",
  adminHost : "localhost",
  tm1Server : "Planning Sample",
  username : "Admin"
}
```

This object has the same format as the response from the login method of `tm1web/api/session/session`.

**adminHost**

Type: String (optional)

Default value: localhost

The admin host from which to retrieve the TM1 server list. If no admin host parameter is specified, the AdminHost parameter value in the `tm1web_config.xml` file is used if it is specified.

**tm1Server**

Type: String (optional)

The TM1 server to log in to.

**adminHostVisible**

Type: Boolean (optional)

Default value: true

If false, the admin host text box is hidden from the login dialog.

**tm1ServersVisible**

Type: Boolean (optional)

Default value: true

If false, the list of TM1 servers are hidden from the login dialog

The `adminHost`, `tm1Server`, `adminHostVisible`, and `tm1ServersVisible` properties can be configured with the `set` method.

For example:

```
loginDialog.set("adminHost", "Planning Sample");
```

**Methods****show()**

Displays the login dialog box.

**destroy()**

Destroys the login dialog box.

For more information, see [Dojo documentation for dijit.\\_WidgetBase](https://dojotoolkit.org/reference-guide/1.10/dijit/_WidgetBase.html) ([https://dojotoolkit.org/reference-guide/1.10/dijit/\\_WidgetBase.html](https://dojotoolkit.org/reference-guide/1.10/dijit/_WidgetBase.html)).

## TM1 Web URL API

---

Use the TM1 Web URL API to include TM1 Web Websheet and CubeViewer objects in any HTML-based document or web page solution.

### TM1 Web URL API overview

The URL API provides a framework for creating URLs that display TM1 Web Websheet and CubeViewer objects in your own custom web pages.

You can use the URL API to include Websheet and CubeViewer objects in any HTML-based solution such as web pages, web applications, and dashboards. The URL API provides access to Websheet and CubeViewer objects by using a special set of URLs and parameters.

**Development tools**

To create a solution with the URL API, you need knowledge of HTML and an optional knowledge of JavaScript.

For testing purposes and simple examples, you can use the URL API right in the address bar of a web browser. To create a solution with the URL API, you can use a simple text editor or any development environment that works with HTML and JavaScript.

The URL API uses HTML inline frames (<iframe> tag) as the primary way to display CubeViewer and Websheet objects in your custom web pages.

## Features

You can assemble URLs that provide the following capabilities in your custom web pages:

- Websheet and CubeViewer
  - Access and display CubeViewer and Websheet objects
  - Set title dimension elements
  - Control properties such as turning the toolbar on or off
- CubeViewer
  - Display in either grid, chart, or grid and chart mode
  - Change chart type
  - Enable/disable auto-recalculation
  - Save the layout of a cube view
  - Recalculate the view
- Websheet
  - Rebuild Active Forms

## Getting started with the TM1 Web URL API

You create a URL by using a base URL and specific TM1 parameters and then pass the completed URL to the TM1 Web server. The completed URL opens and displays a Websheet or CubeViewer object. You can also use the URL API to apply various actions on these objects.

The base URL and the parameters are separated by the hashtag symbol (#) and assembled in the following format:

*BaseUrl#Parameters*

If you want to include multiple parameters in the same URL, separate them with the ampersand symbol (&).

*BaseUrl#Parameter1=value&Parameter2=value&Parameter3=value*

### Web browser address bar example

Copy and paste the following URL into the address bar of your web browser to see a simple example of the URL API.

```
http://localhost:9510/tm1web/
UrlApi.jsp#Action=Open&Type=CubeViewer&Cube=plan_BudgetPlan&View=Budget%20Input
%20Detailed&AccessType=Public&AdminHost=localhost&TM1Server=Planning
%20Sample&Username=admin&Password=apple
```

### Using the URL API in web pages

The URL API uses HTML inline frames (<iframe> tag) to display CubeViewer and Websheet objects in your custom web pages. The <iframe> tag is the primary way to display CubeViewer and Websheet objects with the URL API.

After a TM1 Web object is displayed in an iframe, you can then apply actions on that object by updating the src (source) property of the iframe with a new URL.

For more information, see [“Using HTML iframe tags to display TM1 Web objects”](#) on page 100.

## TM1 Web URL API base URL

Use the base URL as the foundation for building all of your requests with the TM1 Web URL API.

An example of the base URL is shown in the following sample:

```
http://localhost:9510/tm1web/UrlApi.jsp
```

You combine the base URL with one or more parameters to make a complete request.

The base URL uses the following format:

```
http://WebServerName:PortNumber/tm1web/UrlApi.jsp
```

### **WebServerName**

The domain name or IP address of the computer that is hosting the TM1 Web server.

For example, if you are working directly on the computer that is running TM1 Web server, you can use localhost for the *WebServerName* parameter.

```
http://localhost:9510/tm1web/UrlApi.jsp
```

If the TM1Web server is running on a remote computer, use the name of that system as follows:

```
http://MyWebServer:9510/tm1web/UrlApi.jsp
```

```
http://www.example.com:9510/tm1web
```

### **PortNumber**

The port number for the web application server.

The standard TM1 installation uses a port number of 9510.

### **UrlApi.jsp**

The capabilities of the TM1 Web URL API are provided through the `UrlApi.jsp` file.

## TM1 Web URL API parameters

Parameters define which TM1 Web object you want to open and the actions to apply to those objects. You build a complete URL string by adding parameters to the base URL.

The base URL and the parameters are separated by the hashtag symbol (#) and assembled in the following format:

```
BaseUrl#Parameters
```

For example:

```
http://localhost:9510/tm1web/UrlApi.jsp#HideDimensionBar=true
```

If you include more than one parameter, separate them with the ampersand symbol (&).

```
BaseUrl#Parameter1=value&Parameter2=value&Parameter3=value
```

**Note:** Parameters are not case-sensitive. Either "Action" or "action" work the same, however capitalization is recommended for readability.

The most common parameters include Action and Type, which are used to open Workbook and CubeViewer objects. For example, the following URL shows an example of using parameters to open a CubeViewer object.

```
http://localhost:9510/tm1web/UrlApi.jsp#Action=Open&Type=CubeViewer&Cube=plan_BudgetPlan&View=Budget%20Input%20Detailed&AccessType=Public&AdminHost=localhost&TM1Server=Planning%20Sample
```

After you open a Worksheet or CubeViewer object in your web page, you can then use parameters to apply more actions to the object. For example, the following URLs use the AutoRecalc and HideDimensionBar parameters.

```
http://localhost:9510/tm1web/UrlApi.jsp#AutoRecalc=true
```

http://localhost:9510/tm1web/UrlApi.jsp#HideDimensionBar=true

For more information about working with parameters, see the following topics:

- [“Using the Action parameter with TM1 Web objects”](#) on page 103.
- [“Using the Open parameter to open a TM1 Web object”](#) on page 103.
- [“Applying parameters and actions to an existing TM1 Web object”](#) on page 104.

### Using URL escape characters with the URL API

Use URL escape characters when creating URLs that contain spaces or other special characters.

Some common examples of URL escape characters include the following items:

Character	Escape Character
Space	%20
\$	%24
%	%25
&	%26
=	%3D

## TM1 Web URL API concepts

The basic concepts of using the URL API include displaying the objects in HTML iframes, specifying login credentials, opening objects, and applying actions.

### Using HTML iframe tags to display TM1 Web objects

Use HTML inline frames (<iframe> tag) to display CubeViewer and Websheet objects with the URL API in your custom web pages.

The <iframe> tag is the primary way to display CubeViewer and Websheet objects in your custom web pages with the URL API.

After a TM1 Web object is displayed in an iframe, you can then apply actions on that object by updating the src (source) property of the iframe with a new URL.

### Example

The following example uses a standard HTML button and a JavaScript function to load a Websheet into an iframe.

```
<!-- Button to load the websheet -->
<button onClick="loadWebsheet();">Load Websheet</button>

<!-- The iframe to host and display the Websheet -->
<iframe id="websheetId" style="width:100%; height:100%;"></iframe>

<script type="text/javascript">

    // The function to assemble the required URL and display the Websheet
    function loadWebsheet() {

        // Get a reference to the iframe
        webSheet = document.getElementById("websheetId");

        // Assemble the URL that specifies the Websheet you want to open
        baseUrl = "http://localhost:9510/tm1web/UrlApi.jsp";
        var websheetURL = baseUrl + "#Action=Open&Type=WebSheet";
        websheetURL = websheetURL + "&Workbook=Applications/Planning Sample/";
        websheetURL = websheetURL + "Management Reporting/Actual v Budget";
        websheetURL = websheetURL + "&AdminHost=localhost&TM1Server=Planning Sample";

        // Assign the URL to the iframe to display the Websheet
        webSheet.src = websheetURL;
    }
</script>
```

```
};  
</script>
```

### Specifying TM1 Admin Host and TM1 Server parameters with the URL API

You can set the TM1 Admin Host and server name in the URL string by using the `AdminHost` and `TM1Server` parameters.

The `AdminHost` and `TM1Server` parameters can be included in the URL with the `#Action=Open` command or implicitly specified with the use of a session token.

These values are optional in the URL, but must be provided to TM1 in one of the following ways.

- In the `tm1web_config.xml` file
- With a session token
- In the URL string
- Posted to the TM1 Web server using the form-based login
- Provided by the user when prompted by TM1 Web

If these values are not found, then TM1 will prompt the user for this information with a mini pop up window.

The Admin Host and server name are determined in the following order:

1. If a session token is specified, the admin host and TM1 server are determined from that first since it points to a specific session.
2. If the `AdminHost` and `TM1Server` parameters are set in the URL, they will override the values in the `tm1web_config.xml` file.
3. If these values are absent in the URL string, TM1 Web tries to determine if they are set in the `tm1web_config.xml` file.
4. If the `AdminHost` and `TM1Server` parameters are absent from both the URL string and the `tm1web_config.xml` file, then the system prompts the user for this information in a pop up window.

### Example

These parameters use the following format:

```
&AdminHost=AdminHostName&TM1Server=TM1ServerName
```

where:

#### ***AdminHostName***

Name of the system where the TM1 Admin Host is running.

#### ***TM1ServerName***

Name of the TM1 server to log in to.

For example, the following sample code uses the local system and the TM1 Planning Sample database.

```
&AdminHost=localhost&TM1Server=Planning Sample
```

### Managing user login and logout with the URL API

To view TM1 Web objects with the URL API, you must log in to the IBM TM1 server.

You can manage the user login process in the following different ways.

#### **Session token login**

The session token login tracks unique user sessions between multiple TM1 Web instances, TM1 Admin hosts, and TM1 servers.

The session token login is the recommended login approach. Use this login approach if your users log in to multiple instances of TM1 Web or separate TM1 servers at the same time.

For more information, see [“TM1 Web API session login”](#) on page 91.

## TM1 Session ID login

Users can also log in by specifying a TM1 server session with an admin host, TM1 server name, and TM1SessionId. The TM1SessionId corresponds to a user session on a TM1 server. Every TM1 Web session requires a TM1 server session. The TM1 server session that is used by a TM1 Web session will never change and must be generated or specified on creation. Multiple TM1 Web sessions can use the same TM1 server session.

This login method creates a new TM1 Web session and reuses the TM1 server session corresponding to the TM1SessionId. If a TM1 server session is shared between TM1 Web sessions, invalidating the TM1 server session results in the TM1 Web sessions also being invalidated.

A TM1SessionId parameter can be included in the URL to support this type of login authentication. For example:

```
http://localhost:9510/tm1web/UrlApi.jsp#Action=Open&Type=WebSheet&Workbook=Applications/  
Planning Sample/Bottom Up Input/Budget Input&AdminHost=localhost&TM1Server=Planning  
Sample&TM1SessionId=<valid TM1 session ID>
```

## Include user credentials in the URL

You can specify login information in the URL when you access TM1 Web objects. The URL must include values for AdminHost, TM1Server, UserName, or Password.



**CAUTION:** Specifying a password in the URL is not secure.

## Pop-up login window

If all, or some, of the login information is not provided in any other way, then a pop-up window displays to prompt the user to log in before the TM1 Web objects can be displayed.

## Form-based login

You can use a standard HTML form with input fields to collect a user's login credentials and post the information to the TM1 Web server. For more information, see [“TM1 Web URL API form-based login” on page 102](#).

If you are using IBM Cognos Analytics Security authentication, a CamPassport parameter can be specified.

## TM1 Web URL API form-based login

You can use a standard HTML form with input fields to collect a user's login credentials and post the information to the TM1 Web server.

Make sure your form includes `<input>` fields with the following names. The field names and their related values are submitted to the TM1 Web server when you post the form.

- AdminHost
- TM1Server
- Username
- Password

## Example

```
<!-- Login form -->  
<form id="loginInfoForm" method="post">  
  Admin Host: <input type="text" value="localhost" name="AdminHost" /><br>  
  TM1 Server: <input type="text" value="Planning Sample" name="TM1Server" /><br>  
  User Name: <input type="text" value="admin" name="Username" /><br>  
  Password: <input type="password" value="apple" name="Password" /><br>  
  <input type="button" value="Submit" onclick="loadCubeview();" />  
</form>  
  
<!-- The iframe to host and display the TM1 Web object -->  
<iframe id="cubeviewId" name="cubeviewIFrame" style="width:100%; height:100%;"></iframe>  
  
<script type="text/javascript">
```

```

// This function submits the login form and opens a CubeViewer
function loadCubeview() {

    // Get a reference to the login form
    var loginForm = document.getElementById("loginInfoForm");

    var baseUrl = "http://localhost:9510/tm1web/UrlApi.jsp";

    var params = "#Action=Open&Type=CubeViewer&Cube=plan_BudgetPlan";
    params = params + "&View=Budget Input Detailed&AccessType=Public";

    // Assign the URL to the action property of the login form
    loginForm.action = baseUrl + params;

    // NOTE: Be sure to use the iframe name for the target of the login form
    loginForm.target = "cubeviewIFrame";

    // Submit the form to login and display the TM1 Web object
    loginForm.submit();
};
</script>

```

### Logging out from the TM1 Web URL API

Use the Action=Logout parameter to end the current user session with the URL API.

You apply the logout action to an iframe that is already displaying a TM1 Web object. The logout action ends the session that opened that specific TM1 Web object and also ends the session for any other URL API instances under the same session.

The Logout action uses the following format:

`http://localhost:9510/tm1web/UrlApi.jsp#Action=Logout`

### Example

The following example ends the session that is associated with the iframe and the related TM1 Web object.

```

function logout() {

    var baseUrl = "http://localhost:9510/tm1web/UrlApi.jsp";

    var webSheet = document.getElementById("websheetId");
    webSheet.src = baseUrl + "#Action=Logout";
};

```

### Using the Action parameter with TM1 Web objects

The Action parameter specifies the type of action to perform on a TM1 Web object.

The most common action type is the #Action=Open command which can open either a CubeViewer or Websheet object.

Use the Action parameter in the URL string as follows:

`#Action=TypeOfAction`

The *TypeOfAction* value can be one of the supported actions such as Open, Recalc, or Close.

For a complete list of the available action types, see [“URL API Action parameter” on page 112](#).

### Example

For example, the following URL opens a TM1 Web CubeViewer object.

`http://localhost:9510/tm1web/UrlApi.jsp#Action=Open&Type=CubeViewer&Cube=plan_BudgetPlan&View=Budget Input Detailed&AccessType=Public&AdminHost=localhost&TM1Server=Planning Sample`

### Using the Open parameter to open a TM1 Web object

To open and display a TM1 Web object, use the Action=Open command and the Type parameter.

The Open parameter specifies that you want to open and display a TM1 Web object and the Type parameter specifies which type of object.

Action=Open&Type=*object\_type*

The *object\_type* can be either WebSheet or CubeViewer. Depending on the object type, additional parameters are required to specify the exact object to open. You can also set the title selection and other display properties in the same URL when you use the **Open** command.

For example, the following URL shows an example of using the Open and Type parameters to open a CubeViewer object.

```
http://localhost:9510/tm1web/  
UrlApi.jsp#Action=Open&Type=CubeViewer&Cube=plan_BudgetPlan&View=Budget%20Input  
%20Detailed&AccessType=Public&AdminHost=localhost&TM1Server=Planning%20Sample
```

For more information about opening objects, see the following topics:

- [“Displaying Websheet objects with the URL API” on page 105.](#)
- [“Displaying CubeViewer objects with the URL API” on page 107.](#)

After you open a Websheet or CubeViewer object in your web page, you can then use parameters to apply more actions to the object. For more information, see [“Applying parameters and actions to an existing TM1 Web object” on page 104.](#)

### Applying parameters and actions to an existing TM1 Web object

After a TM1 Web object is displayed in your web page, you can then use parameters to apply more actions to that specific object by updating the URL for the object.

To apply more actions to a Websheet or CubeViewer object that is already displayed, create a new URL with the parameters that you want. Then, apply the new URL to the src (source) property of the iframe where the object is displayed.

If the object is already displayed in an iframe, then you need to append only the action parameters to the base URL to create the new URL.

For example, the following URLs append the AutoRecalc and HideDimensionBar parameters to the base URL.

```
http://localhost:9510/tm1web/UrlApi.jsp#AutoRecalc=true
```

```
http://localhost:9510/tm1web/UrlApi.jsp#HideDimensionBar=true
```

### Note:

The AutoRecalc parameter is only applicable to CubeViewer. It is not supported for websheets.

In websheets, auto recalculation is handled by the UseBookRecalcSetting parameter and the setting of the Excel workbook. For more information, see [TM1 Web Configuration Parameters](#).

### Example

The following example shows a JavaScript function that applies an updated URL to the src property of an iframe that is already displaying a CubeViewer object.

```
<!-- Use this iframe to display the CubeViewer (code not shown) -->  
<iframe id="cubeviewId"></iframe>  
  
<script type="text/javascript">  
  // This function updates an existing CubeViewer object  
  function toggleDimensionBar() {  
    // Get a reference to the existing iframe and CubeViewer  
    cubeView = document.getElementById("cubeviewId");  
  
    // Create an updated URL and apply it to the iframe  
    baseUrl = "http://localhost:9510/tm1web/UrlApi.jsp";  
    cubeView.src = baseUrl + "#HideDimensionBar=True";  
  }  
</script>
```

```
};  
</script>
```

## Displaying Websheet objects with the URL API

A websheet is a Microsoft Excel spreadsheet file that contains TM1 data and that you can view in a web browser. You can use the URL API to display a websheet in an HTML iframe and then apply additional actions and parameters to the websheet.

### Opening a Websheet object

To open a Websheet object with the URL API, use the location path to the websheet as organized in the TM1 Application folder.

### Procedure

1. Open TM1 Web and expand the **Applications** node to locate the websheet you want to open.
2. Build a text string that represents the path to the websheet.

Start the path with `Applications/` and separate any sub-folders with the forward slash `/` symbol.

For example: `Applications/My Reports/Report_2014.xls`

3. Set the `Workbook` parameter in your URL to the path that you assembled.

`#Action=Open&Type=WebSheet&Workbook=Applications/My Reports/Report_2014`

4. Combine the parameters with the base URL to make a complete URL request.

### Example

Copy and paste the following URL directly into the address bar of your web browser to see this example.

```
http://localhost:9510/tm1web/  
UrlApi.jsp#Action=Open&Type=WebSheet&Workbook=Applications/Planning%20Sample/  
Management%20Reporting/Actual%20v  
%20Budget&AdminHost=localhost&TM1Server=Planning%20Sample
```

The following JavaScript function loads a websheet into an iframe.

```
function loadWebsheet() {  
    // Get a reference to an existing iframe that has this ID  
    webSheet = document.getElementById("websheetId");  
  
    // Assemble the URL and assign it to the iframe  
    webSheet.src = baseUrl + "#Action=Open&Type=WebSheet  
&Workbook=Applications/Planning Sample/Management Reporting/Actual v Budget  
&AdminHost=localhost&TM1Server=Planning Sample";  
};
```

### Setting display properties for the Websheet object

You can set the display property for the Websheet object by including the related parameter in your URL.

You can use the following parameter to change the display of a Websheet object:

#### HideToolbar

Turns the toolbar on or off. The default is on.

### Examples

Use the following format in your URL to control the display property of a Websheet object.

```
property=value
```

For example, add the following line to your URL to turn off the display of the toolbar.

```
HideToolbar=True
```

### Selecting dimension title elements for Websheet objects

You can set the current elements in a title dimension of a Websheet object for any cell that contains a SUBNM function.

You can specify the dimension by either sheet number, row number, and column number or by dimension name.

You can select the new element by either element name or element index.

### Format and values

Use the following format to specify the dimension by sheet, row, and column number:

```
Title_S#-R#-C#=elementNameOrIndex
```

Use the following format to specify the dimension by dimension name.

```
Title_dimensionName=elementNameOrIndex
```

Use the following parameters:

#### Title\_S#-R#-C#

Specifies the title dimension by sheet number, row number, and column number.

Replace the # symbols with the values for the sheet, row, and column location of the dimension SUBNM cell in the websheet.

#### Title\_dimensionName

Specifies the title dimension by dimension name.

#### elementNameOrIndex

The string value for the name or numeric value for the index of the new title element that you want to select.

If you want to select the new title element by element index, instead of element name, include the UseIndex parameter in the URL as follows:

```
Title_S#-R#-C#=ElementIndexNumber&UseIndex=true
```

### Example

Use the following example to first open a websheet and then change the title element.

1. Copy and paste the following URL directly into the address bar of your web browser to first open the websheet.

```
http://localhost:9510/tm1web/  
UrlApi.jsp#Action=Open&Type=WebSheet&Workbook=Applications/Planning  
%20Sample/Management%20Reporting/Actual%20v  
%20Budget&AdminHost=localhost&TM1Server=Planning%20Sample
```

2. To change the title element, copy and paste the following URL into the same web browser session.

```
http://localhost:9510/tm1web/UrlApi.jsp#Title_S0-R11-C2=Canada
```

3. Copy and paste only the Title\_S#-R#-C# parameter to the end of the base URL to get the similar results.

**Tip:** Only the parameter section of the URL needs to be updated when you use parameters to apply changes. The base URL can remain unchanged.

```
Title_S0-R11-C2=US
```

4. Use the following sample with the UseIndex parameter to select a new title by element index.

```
Title_S0-R11-C2=3&UseIndex=true
```

## Displaying CubeViewer objects with the URL API

The CubeViewer object displays the TM1 cube view in a custom web page. You can use the URL API to display a CubeViewer object in an HTML iframe and then apply additional actions and parameters to the object as needed.

### Opening a CubeViewer object

To identify and open a TM1 Web CubeViewer object, combine the Action=Open command with the Type, Cube, View and AccessType parameters in your URL.

Use the following format to open a CubeViewer object:

```
#Action=Open&Type=CubeViewer&Cube=CubeName&View=ViewName&AccessType=Status
```

where:

- *CubeName* is the name of cube to which the view belongs.
- *ViewName* is the name of cube view.
- *Status* is the public or private status of the cube view. You must include a value of either `Public` or `Private` to correctly identify the specific cube view that you want to open.

Copy and paste the following URL directly into the address bar of your web browser to see this example.

```
http://localhost:9510/tm1web/  
UrlApi.jsp#Action=Open&Type=CubeViewer&Cube=plan_BudgetPlan&View=Budget%20Input  
%20Detailed&AccessType=Public&AdminHost=localhost&TM1Server=Planning%20Sample
```

Use the following JavaScript function to load a CubeViewer into an iframe.

```
function loadCubeview() {  
    // Get a reference to an existing iframe that has this ID  
    cubeView = document.getElementById("cubeviewId");  
  
    // Assemble the URL and assign it to the iframe  
    cubeView.src = baseUrl + "#Action=Open&Type=CubeViewer  
&Cube=plan_BudgetPlan&View=Budget Input Detailed&AccessType=Public";  
};
```

### Setting display properties for the CubeViewer object

You can set the display properties for the CubeViewer object by including any of the related parameters in your URL.

You can use the following parameters to change the display of a CubeViewer object:

#### AutoRecalc

Turns automatic recalculation on or off. The default is off.

#### Note:

The AutoRecalc parameter is only applicable to CubeViewer. It is not supported for websheets. Auto recalculation mode applies to gestures such as pivots, title changes, and zero suppression changes. Auto recalculation mode does not apply to data changes to leaf cells. Leaf cells always turn green when modified.

In CubeViewer, the AutoRecalc parameter serves the same purpose as the **Automatic Recalculation Mode** toolbar button (which does not exist for websheets). When automatic recalculation mode is off (manual recalculation mode), gestures such as pivots, title changes, and zero suppression changes require a recalculation for data to be refreshed.

In websheets, auto recalculation is handled by the `UseBookRecalcSetting` parameter and the setting of the Excel workbook. For more information, see [TM1 Web Configuration Parameters](#).

### HideDimensionBar

Turns the title bar on or off. The default is on.

**Note:** This setting applies to the CubeViewer object only.

### HideToolbar

Turns the toolbar on or off. The default is on.

### Examples

Use the following format in your URL to control the display properties of a CubeViewer object.

```
property=value
```

For example, add the following lines to your URL to change the display properties of the CubeViewer object.

```
AutoRecalc=False
```

```
HideDimensionBar=True
```

```
HideToolbar=True
```

### Selecting title elements for the CubeViewer object

You can set the title elements in a CubeViewer object by adding the title parameter to your URL to specify the dimension and element name.

Use the following format and parameters:

```
Title_DimensionName=ElementNameOrIndex
```

Parameters:

#### **DimensionName**

The name of the title dimension that you want to change.

#### **ElementNameOrIndex**

The element name or the element index of the new title element you want to select.

If you want to select the new title element by element index, instead of element name, include the `UseIndex` parameter in the URL as follows:

```
&Title_DimensionName=ElementIndex&UseIndex=True
```

### Example

Use the following example to first open a CubeViewer and then change the title element.

1. Copy and paste the following URL directly into the address bar of your web browser to first open the CubeViewer.

```
http://localhost:9510/tm1web/  
UrlApi.jsp#Action=Open&Type=CubeViewer&Cube=plan_BudgetPlan&View=Budget  
%20Input%20Detailed&AccessType=Public&AdminHost=localhost&TM1Server=Planning  
%20Sample
```

2. To change the title element, copy and paste the following URL into the address bar of the same web browser session.

```
http://localhost:9510/tm1web/UrlApi.jsp#Title_plan_version=FY 2003 Budget
```

3. Copy and paste only the parameter to the end of the base URL to update the title element.

Title\_plan\_business\_unit=Canada

**Tip:** You only need to update the parameter section of the URL when using parameters to apply changes. The base URL can remain unchanged.

4. Try using the UseIndex parameter to select a new title by element index.

Title\_plan\_business\_unit=7&UseIndex=True

### Displaying charts with the CubeViewer object

Similar to TM1 Web, the CubeViewer object can display TM1 data in grid-only, chart-only, or a combination of grid and chart mode. Use the DisplayMode and ChartType parameters to control the grid and chart display options.

#### Setting grid and chart display options

You can use the DisplayMode parameter to set the display of a CubeViewer object as grid only, chart only, or combined grid and chart.

The DisplayMode parameter uses the following format:

DisplayMode=*value*

The available options include the following values:

- Grid
- Chart
- GridAndChart

#### Example

DisplayMode=**Chart**

DisplayMode=**Grid**

DisplayMode=**GridAndChart**

#### Setting chart type with the URL API

Set the type of chart you want to display for a CubeViewer object by using the ChartType parameter.

The ChartType parameter uses the following format:

ChartType=*ChartName*

where *ChartName* can be the string value for one of the available chart types such as Column or Pie. For a complete list of available chart types, see [“URL API ChartType parameter” on page 114](#).

#### URL example

Copy and paste the following URL directly into the address bar of your web browser to see this example.

http://localhost:9510/tm1web/

UrlApi.jsp#Action=Open&Type=CubeViewer&Cube=plan\_BudgetPlan&View=Budget%20Input%20Detailed&AccessType=Public&AdminHost=localhost&TM1Server=Planning%20Sample&**DisplayMode=GridAndChart&ChartType=Pie**

#### JavaScript example

```
<body>
<select title="Chart Type" onchange="setChartType(this.value);>
  <option></option>
  <option value="Point">Point</option>
  <option value="Bubble">Bubble</option>
  <option value="Line">Line</option>
  <option value="Spline">Spline</option>
  <option value="StepLine">Step Line</option>
  <option value="Bar">Bar</option>
  <option value="StackedBar">Stacked Bar</option  >
```

```

<option value="Column">Column</option>
<option value="StackedColumn">Stacked Column</option>
<option value="Area">Area</option>
<option value="SplineArea">Spline Area</option >
<option value="StackedArea">Stacked Area</option>
<option value="Pie">Pie</option>
<option value="Doughnut">Doughnut</option>
<option value="Range">Range</option >
<option value="SplineRange">Spline Range</option>
</select>

<iframe id="cubeviewId" style="width:100%; height:100%;"></iframe>

<script type="text/javascript">
    function setChartType(value) {
        if(!value) {
            return;
        }

        cubeView = document.getElementById("cubeviewId");
        baseUrl = "http://localhost:9510/tm1web/UrlApi.jsp";
        cubeView.src = baseUrl + "#ChartType=" + value;
    }
};
</script>
</body>

```

## Upgrading older URL API projects to the new TM1 Web URL API

Use this information to upgrade your custom web pages that used the .NET-based TM1 Web URL API to the new Java-based TM1 Web URL API.

As of IBM TM1 version 10.2.0, TM1 Web runs on a Java™-based web application server. TM1 Web version 10.2.0 does not require or use the Microsoft .NET Framework. Because of these changes, the URL API syntax and features are updated.

### Changes in the TM1 Web 10.2.0 environment

Some of the main changes to TM1 Web are summarized in the following list. For more information about installation, configuration, and architecture, see *Planning Analytics Local Installation and Configuration*.

#### New default installation directory for TM1 Web

As of version 10.2.0, the default installation directory for TM1 Web is here:

```
<TM1_install>\webapps\tm1web\
```

#### New default URL for starting TM1 Web

Use the following new, default URL to open TM1 Web version 10.2.0:

```
http://localhost:9510/tm1web/
```

#### New TM1 Web configuration file and parameters

TM1 Web version 10.2.0 uses a new configuration file named `tm1web_config.xml`. This file replaces the `web.config` file from previous TM1 Web versions.

The new configuration file is location here:

```
<TM1_install>\webapps\tm1web\web-inf\configuration
```

### Changes in the TM1 Web 10.2.2 URL API

The TM1 Web 10.2.2 URL API includes the following changes and updates:

#### Objects

- The TM1 Web Navigation tree object is not supported in the 10.2.2 URL API.
- The 10.2.2 URL API does not use the `ObjectId` parameter to track and apply actions on existing objects in your web page. Instead, the new URL API maintains the current state of the object internally for improved cross-domain usage. You can now apply additional actions on a TM1 Web object by using the `iframe` where the object is displayed.

## Parameters

- Parameters are now separated from the base URL with the hash tag symbol (#) instead of the question mark (?).  
For example: `http://localhost:9510/tm1web/UrlApi.jsp#Parameters`
- The `OpenObject` parameter is renamed to `Open`.
- Parameter values of `Yes` and `No` are replaced with `True` and `False`. Values of `0` and `1` still work.
- The behavior of the `Action=Save` parameter in 10.2.2 is different and applies only to the `CubeViewer` object. This action saves only the layout of the view, and does not save changes to the data. Use the `Recalc` action to save data in a `CubeViewer` object.
- The `HideTitlebar` parameter is renamed to `HideDimensionBar`.
- The `HideTabs` parameter is no longer used.
- The `ChartType` parameter now uses string values instead of numeric values.

## Required code changes for updating to the 10.2.2 URL API

To upgrade projects to the new URL API, review and apply the following code changes.

### Change the base URL

Change your existing base URLs to use the new format for TM1 Web 10.2.2.

- Replace this URL: `http://HostName/TM1Web/TM1WebMain.aspx`
- With this URL: `http://HostName:9510/tm1web/UrlApi.jsp`

The `UrlApi.jsp` file replaces the `TM1WebMain.aspx` handler file.

### Update the URL parameters

Review the list of changes in the TM1 Web 10.2.2 URL API.

For example, parameters are now separated from the base URL with the hash tag symbol (#) and some parameters are renamed.

### Update the login process

The 10.2.2 URL uses a new session token login approach for uniquely identifying login sessions. A new form-based login is also available.

### Replace the `ObjectId` parameter

Update your code anywhere you used the `ObjectId` parameter to track objects that you opened.

Instead, the new URL API maintains the current state of the object internally for improved cross-domain usage. Use this feature to apply more actions on a TM1 Web object by updating the `src` property of the `iframe` whenever you want to update an object.

## TM1 Web URL API parameter reference

Use parameters to define which IBM TM1 Web object you want to open and the actions to perform on that object. You build a complete URL string by adding parameters to the base URL.

**Note:** Parameter format is show as `&<parameter>=<value>`. In examples, the parameter might appear as `#<parameter>`. The `&` is used to separate parameters, while `#` is used to mark the beginning of the parameters in examples.

### URL API `AccessType` parameter

The `AccessType` parameter specifies the public or private status of the cube view that you want to display.

This parameter is used in combination the `Action` parameter when you open a `CubeViewer` object.

### Format

`&AccessType=Value`

## Values

Value	Description
Private	Specifies that the cube view has a private status.
Public	Specifies that the cube view has a public status.

## Example

```
function loadCubeview() {
    cubeView = document.getElementById("cubeviewId");

    cubeView.src = baseUrl + "#Action=Open&Type=CubeViewer
    &Cube=plan_BudgetPlan&View=Budget Input Detailed
    &AccessType=Public
    &AdminHost=localhost&TM1Server=Planning Sample";
};
```

## URL API Action parameter

The Action parameter specifies the type of action to perform on an IBM TM1 Web object.

## Format

*&Action=Type\_Of\_Action*

## Values

Value	Description
Close	Closes an existing object.
Logout	Ends the session for any other URL API instances under the same session.
Open	Opens a TM1 Web object.
Rebuild	Recalculates all values and rebuilds all subsets for a TM1 Active Form contained in a websheet.  This action performs the same action as when you click the Rebuild button on the TM1 Web toolbar.
Recalc	Recalculates an existing Websheet or CubeViewer object.
Reload	Reloads the CubeViewer object only.
Save	Saves the layout of a cube view. Applies only to CubeViewer objects.  <b>Note:</b> The Save action does not save any changes to the data in the view. Use the Recalc action to save changed data.

## URL Example

The following URL examples show some actions to perform on a CubeViewer or Websheet object that is already displayed in a web page.

<http://localhost:9510/tm1web/UrlApi.jsp#Action=Save>

<http://localhost:9510/tm1web/UrlApi.jsp#Action=Reset>

<http://localhost:9510/tm1web/UrlApi.jsp#Action=Close>

## JavaScript Example

The following example shows a collection of JavaScript functions that each perform a different action on a CubeViewer or Websheet object.

```
<script type="text/javascript">
  function loadWebsheet() {
    webSheet = document.getElementById("websheetId");

    webSheet.src = baseUrl + "#Action=Open&Type=WebSheet
    &Workbook=Applications/Planning Sample/Management Reporting/Actual v Budget
    &AdminHost=localhost&TM1Server=Planning Sample";
  };

  function loadCubeview() {
    cubeView = document.getElementById("cubeviewId");

    cubeView.src = baseUrl + "#Action=Open&Type=CubeViewer&Cube=plan_BudgetPlan
    &View=Budget Input Detailed&AccessType=Public
    &AdminHost=localhost&TM1Server=Planning Sample";
  };

  function rebuildActiveForms() {
    webSheet.src = baseUrl + "#Action=Rebuild";
  };

  function recalculate() {
    getActiveIFrame().src = baseUrl + "#Action=Recalc";
  };

  function resetView() {
    cubeView.src = baseUrl + "#Action=Reset";
  };

  function saveView() {
    cubeView.src = baseUrl + "#Action=Save";
  };

  function close() {
    getActiveIFrame().src = baseUrl + "#Action=Close";
  };
</script>
```

### URL API AdminHost parameter

The AdminHost parameter defines the name of the system where the IBM TM1 Admin Host is running. The default value is localhost.

### Format

&AdminHost=*admin\_host\_name*

### Values

The value of the AdminHost parameter is the name of the system where the TM1 Admin server is running.

### Example

```
function loadCubeview() {
  cubeView = document.getElementById("cubeviewId");

  cubeView.src = baseUrl + "#Action=Open&Type=CubeViewer
  &Cube=plan_BudgetPlan&View=Budget Input Detailed&AccessType=Public
  &AdminHost=localhost&TM1Server=Planning Sample";
};
```

### URL API AutoRecalc parameter

Use the AutoRecalc parameter to turn automatic recalculation on or off. The default is off.

#### Format

&AutoRecalc=*value*

#### Values

Value	Description
0, false	Disables automatic recalculation.
1, true	Enables automatic recalculation.

#### Example

```
function toggleAutoRecalcMode(enabled) {  
    getActiveIFrame().src = baseUrl + "#AutoRecalc=" + enabled;  
};
```

### URL API ChartType parameter

Use the ChartType parameter to set the type of chart that you want to display.

#### Format

&ChartType=*chart\_type*

#### Values

Value	Chart Type
Point	Point
Bubble	Bubble
Line	Line
Spline	Spline
Stepline	Step Line
Bar	Bar
Stackedbar	Stacked Bar
Column	Column
Stackedcolumn	Stacked Column
Area	Area
Splinearea	Spline Area
Stackedarea	Stacked Area
Pie	Pie
Doughnut	Doughnut
Range	Range
Splinerange	Spline Range

## Example

```
function setChartType(value) {
    if(!value) {
        return;
    }

    cubeView.src = baseUrl + "#ChartType=" + value;
};
```

## URL API Cube parameter

Use the Cube parameter to specify the name of cube to which the view belongs.

### Format

&Cube=*cube\_name*

### Values

The value of the Cube parameter is the name of the cube, which contains the view that you want to open.

## Example

```
http://localhost:9510/tm1web/UrlApi.jsp#Action=Open&Type=CubeViewer
&Cube=plan_BudgetPlan&View=Budget%20Input%20Detailed&AccessType=Public
&AdminHost=localhost&TM1Server=Planning%20Sample&DisplayMode=GridAndChart
&ChartType=Pie
```

## URL API DisplayMode parameter

Use the DisplayMode parameter to display a CubeViewer object in grid, chart, or grid and chart mode.

### Format

&DisplayMode=*display\_type*

### Values

Value	Description
Chart	Displays the CubeViewer object in chart-only mode.
Grid	Displays the CubeViewer object in grid-only mode.
GridAndChart	Displays the CubeViewer object with both a grid and chart.

## Example

The following example shows a URL to apply to a CubeViewer object that is already displayed.

```
http://localhost:9510/tm1web/UrlApi.jsp#DisplayMode=Chart
```

The following example uses a JavaScript function to change the display mode.

```
function setDisplayMode(value) {
    if(!value) {
        return;
    }

    cubeView.src = baseUrl + "#DisplayMode=" + value;
};
```

### URL API HideDimensionBar parameter

Use the HideDimensionBar parameter to control the display of the dimension title bar for the CubeViewer object. This setting applies to the CubeViewer object only.

#### Format

&HideDimensionBar=*value*

#### Values

Value	Description
1, true	Hides the dimension bar.
0, false	Displays the dimension bar.

#### Example

```
#HideDimensionBar=true
```

### URL API HideToolbar parameter

Use the HideToolbar parameter to control the display of the toolbar for CubeViewer and Websheet objects.

#### Format

&HideToolbar=*value*

#### Values

Value	Description
1, false	Hides the toolbar.
0, true	Displays the toolbar.

#### Example

```
#HideToolbar=1
```

### URL API TM1Server parameter

The TM1Server parameter specifies the IBM TM1 server to log into.

#### Format

&TM1Server=*TM1\_server\_name*

#### Values

The value of the TM1Server parameter is the name of the TM1 server to log in to.

#### Example

```
&TM1Server=Planning Sample
```

### URL API TM1SessionId parameter

The TM1SessionId parameter specifies the IBM TM1 server to log into.

#### Format

`&TM1SessionId=valid_TM1_session_ID`

#### Values

Users can log in by specifying a TM1 server session with an admin host, TM1 server name, and TM1SessionId. The TM1SessionId parameter corresponds to a user session on a TM1 server.

For more information, see [“TM1 Web API session login”](#) on page 91.

#### Example

```
http://localhost:9510/tm1web/UrlApi.jsp#Action=Open&Type=WebSheet&Workbook=Applications/
Planning Sample/Bottom Up Input/Budget Input&AdminHost=localhost&TM1Server=Planning
Sample&TM1SessionId=<valid TM1 session ID>
```

### URL API Type parameter

The Type parameter is used with the Action parameter to specify the type of object that you want to open.

#### Format

`&Type=object_type`

#### Values

Value	Description
CubeViewer	Defines the object as a CubeViewer.
Websheet	Defines the object as a Websheet.

#### Example

```
http://localhost:9510/tm1web/UrlApi.jsp#Action=Open&Type=CubeViewer
&Cube=plan_BudgetPlan&View=Budget%20Input%20Detailed&AccessType=Public
&AdminHost=localhost&TM1Server=Planning%20Sample
```

### URL API View parameter

Use the View parameter to specify the name of the cube view that you want to open.

#### Format

`&View=view_name`

#### Values

The value of the View parameter is the name of the cube view.

#### Example

```
View=Budget%20Input%20Detailed
```

A complete URL is shown in the following example.

```
http://localhost:9510/tm1web/UrlApi.jsp#Action=Open&Type=CubeViewer
&Cube=plan_BudgetPlan&View=Budget%20Input%20Detailed&AccessType=Public
&AdminHost=localhost&TM1Server=Planning%20Sample
```

### URL API Workbook parameter

The Workbook parameter specifies the path in the IBM TM1 server tree of the workbook to be loaded.

### Format

`&Workbook=`*path\_to\_workbook*

### Values

The value of the Workbook parameter is the path to the TM1 Worksheet as organized in the TM1 Application folder.

### Example

`&Workbook=Applications/Planning Sample/Management Reporting/Actual v Budget`

A complete URL is shown in the following example.

```
http://localhost:9510/tm1web/
UrlApi.jsp#Action=Open&Type=WebSheet&Workbook=Applications/Planning%20Sample/
Management%20Reporting/Actual%20v
%20Budget&AdminHost=localhost&TM1Server=Planning%20Sample
```

## TM1 Web JavaScript library

---

You can use the TM1 Web JavaScript library to programmatically access TM1 Web Worksheet and CubeViewer objects in a combined HTML, JavaScript, and Dojo web page development environment. A working knowledge of JavaScript, Dojo Toolkit, and the HTML Document Object Model (DOM) is required for using the JavaScript library.

### Overview

The TM1 Web JavaScript library includes the following main classes:

#### Workbook class

Represents a TM1 Web Worksheet.

#### CubeViewer class

Represents a TM1 Web CubeViewer.

These main classes extend the Dojo Toolkit widget class called `dijit._WidgetBase`. This extension allows the Workbook and CubeViewer objects to be assigned as children of other Dojo objects such as a Dojo tab container or other container.

For more information about Dojo, see the Dojo documentation: <http://dojotoolkit.org/documentation/>.

The Worksheet and CubeViewer objects also have a set of related properties and methods that you can access programmatically. These objects are loaded asynchronously and must finish loading before your code can interact with the objects.

### Note:

In the TM1 Web JavaScript library, the following objects are deprecated:

- `tm1web/cubeview/CubeViewer`
- `tm1web/worksheet/Workbook`

You should use `tm1web/api/CubeViewer` and `tm1web/api/Workbook` instead. The modules within the `tm1web/cubeview` and `tm1web/websheet` packages are now aliases of the modules within the `tm1web/api` package.

## Configuration

The following configuration is required to use the TM1 Web JavaScript library.

1. Install TM1 Web and verify that you can log in to the standard user interface with a web browser.
2. Add the required references to the head section of your custom web page files that use the JavaScript library.

For more information, see [“Required HTML <head> and <body> tags to use the JavaScript library” on page 119.](#)

## Getting started with JavaScript library

After you configured your TM1 Web environment, you can start coding your web pages to access objects with the JavaScript library. For more information and examples, see the following topics:

- [“Loading websheet objects with the JavaScript library” on page 122.](#)
- [“Loading CubeViewer objects with the JavaScript library” on page 123.](#)

## Configuring the AMD loader for the JavaScript Library

As of IBM Planning Analytics Local 2.0.0, it is no longer mandatory to use the version of Dojo that is provided with TM1 Web to load the TM1 Web JavaScript Library modules.

TM1 Web now supports using the AMD loader from Dojo version 1.7 and later to load the JavaScript Library modules.

For more information, see [“Configuring the AMD loader for the JavaScript Library” on page 120.](#)

## Required HTML <head> and <body> tags to use the JavaScript library

The HTML <head> and <body> sections in each custom web page that uses the TM1 JavaScript library must include a set of required tags and references.

Add the following references to any of your HTML documents that use the JavaScript library.

- Include an HTML 5 DOCTYPE declaration.
- Add the meta references to the <head> section.
- Add the class reference to the <body> section.
- Add additional code to handle configuration of the AMD loader to find the JS Library modules correctly.

These references point to files contained under the TM1 Web installation directory.

*TM1\_Installation\_Location\webapps\tm1web\...*

## Example

Use the following tags and references as a template.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
</head>
<body class="claro tm1web"></body>
</html>
```

## Configuring the AMD loader for the JavaScript Library

You can use the AMD loader from Dojo version 1.7 and later to load the JavaScript Library modules.

Before any JavaScript Library module can be imported by using the AMD `require` function, the AMD loader must be configured to find and map the modules. The following example demonstrates how to configure the AMD loader for supported versions of Dojo.

**Note:** In the following examples, `location/to/tm1web/scripts/tm1web` represents the TM1 Web URI. An example of this location might be `http://localhost:9510/tm1web/scripts/tm1web`.

The following example shows how to configure the AMD loader for Dojo versions 1.8, 1.9 and 1.10.

```
require({
  packages: [
    {
      name: "tm1web",
      location: "location/to/tm1web/scripts/tm1web"
    },
    {
      name: "tm1webCom",
      location: "location/to/tm1web/scripts/com"
    },
    {
      name: "tm1webDojo",
      location: "location/to/tm1web/scripts/dojo"
    },
    {
      name: "tm1webDijit",
      location: "location/to/tm1web/scripts/dijit"
    },
    {
      name: "tm1webDojox",
      location: "location/to/tm1web/scripts/dojox"
    }
  ],
  map: {
    tm1web: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webCom: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webRave: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webDojo: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webDijit: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webDojox: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    }
  }
});
```

The following example shows how to configure the AMD loader for Dojo 1.7.

```
require({
  packages: [
    {
      name: "tm1web",
      location: "location/to/tm1web/scripts/tm1web",
      packageMap: {
        dojo: "tm1webDojo",
        dijit: "tm1webDijit",
        dojox: "tm1webDojox",
        com: "tm1webCom"
      }
    },
    {
      name: "tm1webCom",
      location: "location/to/tm1web/scripts/com",
      packageMap: {
        dojo: "tm1webDojo",
        dijit: "tm1webDijit",
        dojox: "tm1webDojox",
        com: "tm1webCom"
      }
    },
    {
      name: "tm1webRave",
      location: "location/to/tm1web/scripts/com",
      packageMap: {
        dojo: "tm1webDojo",
        dijit: "tm1webDijit",
        dojox: "tm1webDojox",
        com: "tm1webCom"
      }
    },
    {
      name: "tm1webDojo",
      location: "location/to/tm1web/scripts/dojo",
      packageMap: {
        dojo: "tm1webDojo",
        dijit: "tm1webDijit",
        dojox: "tm1webDojox"
      }
    },
    {
      name: "tm1webDijit",
      location: "location/to/tm1web/scripts/dijit",
      packageMap: {
        dojo: "tm1webDojo",
        dijit: "tm1webDijit",
        dojox: "tm1webDojox"
      }
    },
    {
      name: "tm1webDojox",
      location: "location/to/tm1web/scripts/dojox",
      packageMap: {
        dojo: "tm1webDojo",
        dijit: "tm1webDijit",
        dojox: "tm1webDojox"
      }
    }
  ]
});
```

The following example shows a complete configuration.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<script src="path/to/the/1.10/version/of/dojo.js"></script>
<script>
  require({
    packages: [
      {
        name: "tm1web",
        location: "http://localhost:9510/tm1web/scripts/tm1web"
      },
    ],
  });
</script>
```

```

    {
      name: "tm1webCom",
      location: "http://localhost:9510/tm1web/scripts/com"
    },
    {
      name: "tm1webDojo",
      location: "http://localhost:9510/tm1web/scripts/dojo"
    },
    {
      name: "tm1webDijit",
      location: "http://localhost:9510/tm1web/scripts/dijit"
    },
    {
      name: "tm1webDojox",
      location: "http://localhost:9510/tm1web/scripts/dojox"
    }
  ],
  map: {
    tm1web: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webCom: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webRave: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webDojo: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webDijit: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    },
    tm1webDojox: {
      dojo: "tm1webDojo",
      dijit: "tm1webDijit",
      dojox: "tm1webDojox",
      com: "tm1webCom"
    }
  }
});

require([
  "tm1web/api/Workbook"
], function(Workbook) {
  // Create and work with Workbook object
});
</script>
</head>
<body class="claro tm1web"></body>
</html>

```

## Loading worksheet objects with the JavaScript library

Use JavaScript to instantiate a worksheet object. After the object is loaded, you can then assign it as a descendant of the document body to display it in your web page.

You load a worksheet object by using the following format to specify the required properties and optional functions that define the object.

```
new Workbook({properties ..., functions ...});
```

The *properties* include values that specify the login credentials and the websheet object that you want to open.

The *functions* can include optional code to notify you about onLoad and onTitleDimensionElementChange events for the object.

For more information, see [“TM1 Web JavaScript library Workbook class” on page 128.](#)

### Example

The following example shows a JavaScript function that loads a websheet object.

The code to instantiate the object must use the specific AMD (Asynchronous Module Definition) syntax and the AMD require keyword. After the object is created, the function assigns it as the child of a document body.

```
// Load websheet with parameters for adminHost, tm1Server, username and password
function loadWebsheet() {
  require([
    "tm1web/api/Workbook"
  ], function(Workbook){
    var loadedWebsheet = new Workbook({
      adminHost: "localhost",
      tm1Server: "Planning Sample",
      username: "admin",
      password: "apple",
      path: "Applications/Planning Sample/Management Reporting/Actual v Budget",
      onLoad: function() {
        console.debug("Workbook loaded successfully.");
      }
    });
    // Add websheet to the document body
    document.body.appendChild(loadedWebsheet.domNode);
    loadedWebsheet.startup();
  });
};
```

The following example loads a websheet object by using a session token for the login.

```
// Load websheet with a session token
function loadWebsheet() {
  require([
    "tm1web/api/Workbook"
  ], function(Workbook){
    var loadedWebsheet = new Workbook({
      sessionToken: "yourSessionToken",
      path: "Applications/Planning Sample/Management Reporting/Actual v Budget",
      onLoad: function() {
        console.debug("Workbook loaded successfully.");
      }
    });
    // Add websheet to the document body
    document.body.appendChild(loadedWebsheet.domNode);
    loadedWebsheet.startup();
  });
};
```

## Loading CubeViewer objects with the JavaScript library

Use JavaScript to instantiate a CubeViewer object. After the object is created, you can then assign it as a descendant of the document body to display it in your web page.

You load a CubeViewer object by using the following format to specify the required properties and optional functions that define the object.

```
new CubeViewer({properties ..., functions ...});
```

The *properties* include values that specify the login credentials and the CubeViewer object that you want to open.

The *functions* can include optional code to notify you about onLoad and onTitleDimensionElementChange events for the object.

For more information, see [“TM1 Web JavaScript library CubeViewer class” on page 136](#).

### Example

The following example shows a JavaScript function that loads a CubeViewer object.

The code to instantiate the object must use the specific AMD syntax and the Dojo require keyword. After the object is created, the function assigns it as the child of a document body.

```
function loadCubeview() {
  require([
    "tm1web/api/CubeViewer",
  ], function(CubeViewer) {
    var loadedCubeview = new CubeViewer({
      adminHost: "localhost",
      tm1Server: "Planning Sample",
      cube: "plan_BudgetPlan",
      view: "Budget Input Detailed",
      isPublic: true,
      onLoad: function() {
        console.debug("CubeViewer loaded successfully.");
      }
    });

    // Add cubeview to the document body
    document.body.appendChild(loadedCubeview.domNode);

    loadedCubeview.startup();
  });
};
```

The following example loads a CubeViewer object by using a session token for the login.

```
function loadCubeview() {
  require([
    "tm1web/api/CubeViewer",
  ], function(CubeViewer) {
    var loadedCubeview = new CubeViewer({
      sessionToken: "yourSessionToken",
      cube: "plan_BudgetPlan",
      view: "Budget Input Detailed",
      isPublic: true,
      onLoad: function() {
        console.debug("CubeViewer loaded successfully.");
      }
    });

    // Add cubeview to the document body
    document.body.appendChild(loadedCubeview.domNode);

    loadedCubeview.startup();
  });
};
```

## JavaScript library callback functions

You can define a callback function when you instantiate Websheet and CubeViewer objects. The callback function traps changes to the title dimensions in the related object so you can process the event.

Websheet and CubeViewer objects both use the same format for defining a callback function. You add the callback function directly within the function that instantiates the TM1 Web object. Your code to handle the event goes inside this function.

## Format

The callback function is defined with the following format:

```
onTitleDimensionElementChange: function(elementInfo) {  
    // Add your code here to handle the title change event  
}
```

When a change to a title dimension is detected, the `elementInfo` object is passed to the callback function. The content of `elementInfo` is different for Websheet and CubeViewer objects. Use this information to see which dimension title has changed.

### Websheet `elementInfo` object:

#### **sheetIndex**

Type: Integer

The zero-based index of the sheet that contains the SUBNM cell that was changed.

#### **rowIndex**

Type: Integer

The zero-based index of the row that contains the SUBNM cell that was changed.

#### **columnIndex**

Type: Integer

The zero-based index of the column that contains the SUBNM cell that was changed.

#### **dimension**

Type: String

The name of the dimension.

#### **element**

Type: String

The name of the element.

#### **elementIndex**

Type: Integer

The one-based index of the dimension element.

### CubeViewer `elementInfo` object:

#### **dimension**

Type: String

The name of the dimension.

#### **element**

Type: String

The name of the element.

#### **elementIndex**

Type: Integer

The one-based index of the dimension element.

### Websheet callback function example

The following example shows a callback function that is defined within the function that loads a Websheet object.

```
function loadWebsheet() {  
    require([  
        "tm1web/api/Workbook"  
    ], function(Workbook){  
        var loadedWebsheet = new Workbook({  
            sessionToken: "yourSessionToken",  
        });  
    });  
}
```

```

        path: "Applications/Planning Sample/Management Reporting/Actual v Budget",,
        onLoad: function() {
            console.debug("Workbook loaded successfully.");
        },
        onTitleDimensionElementChange: function(elementInfo) {
            console.debug("Title dimension element changed:");
            console.debug(elementInfo);
        }
    });

    document.body.appendChild(loadedWebsheet.domNode);

    loadedWebsheet.startup();
});
};

```

### CubeViewer callback function example

The following example shows a callback function that is defined within the function that loads a CubeViewer object.

```

function loadCubeview() {
    require([
        "tm1web/api/CubeViewer"
    ], function(CubeViewer) {
        var loadedCubeview = new CubeViewer({
            sessionToken: "yourSessionToken",
            cube: "plan_BudgetPlan",
            view: "Budget Input Detailed",
            isPublic: true,
            onLoad: function() {
                console.debug("CubeViewer loaded successfully.");
            },
            onTitleDimensionElementChange: function(elementInfo) {
                console.debug("Title dimension element changed:");
                console.debug(elementInfo);
            }
        });
        document.body.appendChild(loadedCubeview.domNode);

        loadedCubeview.startup();
    });
};

```

## JavaScript library sample code for properties and methods

After you load Websheet and CubeViewer objects with the TM1 Web JavaScript library, you can then apply the available properties and methods to them by using an object oriented approach.

The following code samples show how to apply different properties and methods.

### Websheet object

- Rebuild the Active Forms in a Websheet
- Recalculate a Websheet

### CubeViewer object

- Turn on/off auto recalculation mode
- Turn on/off the dimension title bar
- Reset a CubeViewer object to its original view
- Save a view
- Set the display mode and chart type

### Websheet and CubeViewer objects

- Close a Websheet or CubeViewer object
- Logout

## Example

```
<script type="text/javascript">

    // Rebuild the active form in a Websheet
    // -----
    function rebuildActiveForms() {
        loadedWebsheet.rebuildActiveForms().then(
            function() {
                console.debug("Active form rebuild completed.");
            },
            function(message) {
                console.error(message);
            }
        );
    };

    // Recalculate a Websheet
    // -----
    function recalculate() {
        loadedWebsheet.recalculate().then(
            function() {
                console.debug("Recalculate completed successfully.");
            },
            function(message) {
                console.error(message);
            }
        );
    };

    // Set the AutoRecalcMode for a CubeViewer object
    // -----
    function toggleAutoRecalcMode(enabled) {
        loadedCubeview.set("automaticRecalculation", enabled).then(
            function() {
                var message = enabled ?
                    "Enabling auto recalc completed successfully." :
                    "Disabling auto recalc completed successfully.";
                console.debug(message);
            },
            function(message) {
                console.error(message);
            }
        );
    };

    // Turn on/off the dimension title bar for a CubeViewer object
    // -----
    function toggleDimensionBar(visible) {
        loadedCubeview.set("dimensionBarVisible", visible);
    };

    // Reset a CubeViewer object to it's original view
    // -----
    function resetView() {
        loadedCubeview.reset().then(
            function() {
                console.debug("View reset completed successfully.");
            },
            function(message) {
                console.error(message);
            }
        );
    };

    // Save a view for a CubeViewer object
    // -----
    function saveView() {
        loadedCubeview.save().then(
            function() {
                console.debug("Saving view completed successfully.");
            },
            function(message) {
                console.error(message);
            }
        );
    };

    // Close a Websheet or CubeViewer object
    // -----

```

```

function close() {
    loadedWebsheet.destroy();
};

// Set the display mode for a CubeViewer object
// Valid values include Grid, Chart, GridAndChart
// -----
function setDisplayMode() {

    require(["tm1web/cubeview/DisplayMode"], function(DisplayMode) {
        loadedCubeview.set("displayMode", DisplayMode.Grid).then(
            function() {
                console.debug("Display mode change completed successfully.");
            },
            function(message) {
                console.error(message);
            }
        );
    });
};

// Set the chart type for a CubeViewer object
// -----
function setChartType() {

    require(["tm1web/cubeview/ChartType"], function(ChartType) {
        loadedCubeview.set("chartType", ChartType.Pie).then(
            function() {
                console.debug("Chart type change completed successfully.");
            },
            function(message) {
                console.error(message);
            }
        );
    });
};

// Logout from the session associated with the specified TM1 Web object
// -----
function logout() {
    loadedCubeview.logout().then(
        function() {
            console.debug("Session destroyed.");
        },
        function(message) {
            console.error(message);
        }
    );
};

</script>

```

## TM1 Web JavaScript library Workbook class

The Workbook class represents a TM1 Web Websheet object.

Workbook objects extend the Dojo widget object (`dijit._WidgetBase`) and can be assigned as a child object of a Dojo tab container (`dijit.layout.TabContainer`) or other container. For more information, see [Dojo documentation](http://dojotoolkit.org/documentation/) (<http://dojotoolkit.org/documentation/>).

In addition to the available properties and methods of the Dojo widget object, Workbook objects also have TM1 related properties and methods that you can access programmatically.

Workbook objects are loaded asynchronously and must finish loading before your code can interact with the objects.

### Format

You load a Websheet object by using the following format to specify the required properties and optional functions that define the object.

```
new Workbook({properties ..., functions ...});
```

### Properties

The *properties* include the following values that define the Websheet object.

- adminHost
- tm1Server
- username
- password
- camPassport
- sessionToken
- objectId
- path

**Note:** You can provide login credentials as either a session token and an object ID, or by including separate values for TM1 Admin host, TM1 Server, user name, password, or camPassport.

## Functions

The *functions* can include the following optional code:

- Use the onLoad function so that you can be notified when the object is loaded and ready to interact with.
- Use the onTitleDimensionElementChange declaration so that you can process the event when a user changes a dimension title in the related object.
- Use the OnActionButtonExecution declaration so that you can process the event when an action button is executed.

## Example

The following example shows a JavaScript function that loads a Websheet object.

The login credentials can be provided by using a session token.

**Note:** The Workbook class accepts objectId as a parameter during construction. The objectId should be included with a sessionToken to identify the TM1 Web session.

```
// Load Websheet with a session token
function loadWebsheet() {
  require([
    "tm1web/api/Workbook"
  ], function(Workbook){
    var loadedWebsheet = new Workbook({
      sessionToken: "yourSessionToken",
      objectId: "objectIdOfNewWorkbook"
    }, {
      onLoad: function() {
        console.debug("Workbook loaded successfully.");
      }
    });

    // Add websheet to the document body
    document.body.appendChild(loadedWebsheet.domNode);

    loadedWebsheet.startup();
  });
};
```

## Workbook properties

This Workbook class has the following properties.

When instantiating either a CubeViewer or Workbook, the following properties are common between the two objects.

### sessionToken

Type: String

Specifies the TM1 Web session to use for this object. Do not use this property with the properties for `adminHost`, `tm1Server`, `username`, `password`, and `camPassport`. If this property is not specified, and no additional credentials are provided, the user is prompted with a login dialog during startup.

### **objectId**

Type: String

The ID of the Workbook. A unique identifier that you can use to reference the specific Workbook.

The `objectId` should be included with a `sessionToken` to identify the TM1 Web session.

For example:

```
new Workbook({
  sessionToken: "previousSessionToken",
  objectId: "objectIdOfNewWorkbook"
});
```

### **adminHost**

Type: String

Default: localhost

The admin host to use when the object is loaded. Do not use this property with the `sessionToken` property.

### **tm1Server**

Type: String

The TM1 server to use when the object is loaded. Do not use this property with the `sessionToken` property. If unspecified and no `sessionToken` is provided, the user is prompted with a login dialog during startup.

### **username**

Type: String

The user name to use when the object is loaded. Do not use this property with the `sessionToken` or `camPassport` properties. If unspecified and no `sessionToken` or `camPassport` is provided, the user is prompted with a login dialog during startup.

### **password**

Type: String

The password to use when the object is loaded. If unspecified and no `sessionToken` is provided, the user is prompted with a login dialog during startup.

### **camPassport**

Type: String

The BI authentication passport (CAM passport) to use when you load an object. Do not use this property with `username` or `sessionToken`.

### **domNode**

Type: HTMLElement

The underlying HTML element that represents the widget. This property is automatically defined during object construction and should not be provided during instantiation.

For more information, see [Dojo documentation for dijit.\\_WidgetBase](https://dojotoolkit.org/reference-guide/1.10/dijit/_WidgetBase.html) ([https://dojotoolkit.org/reference-guide/1.10/dijit/\\_WidgetBase.html](https://dojotoolkit.org/reference-guide/1.10/dijit/_WidgetBase.html)).

The following properties are used when you instantiate a `Workbook` object only.

### **path**

Type: String

The path in the TM1 server application folder tree for the workbook to be loaded.

For example: "Applications/Planning Sample/Bottom Up Input/Budget Input"

### **replaceOnNavigate**

Type: Boolean (default true)

If `true`, during action button navigation to a new workbook, this widget will be replaced with the new workbook and the existing workbook will be closed.

If `false`, it is the consumer's responsibility to create a new workbook or replace this one using the information provided to the `onActionButtonExecution` method.

### Get properties

All properties that get a value are called with the following format:

```
get("property_Name").
```

For example: `get("sandboxes");`

### sandboxes

Retrieves all available sandboxes.

Returns `dojo.promise.Promise` as a promise that is resolved when the sandboxes are retrieved. When the promise is resolved, an Array of objects that represent the available sandboxes is passed to any callback registered with the promise.

Each object uses the following format:

#### name

(String) - The name of the sandbox.

#### active

(Boolean) - `True` if this sandbox is the active sandbox for the object, else `false`.

#### baseSandbox

(Boolean) - `True` if this sandbox is the base sandbox, else `false`.

#### defaultSandbox

(Boolean) - `True` if this sandbox is the default sandbox, else `false`.

### Set properties

All properties that set a value are called with the following format:

```
set("property_Name", value)
```

For example: `set("activeSandbox", "theSandbox");`

### activeSandbox

Sets the specified sandbox as active.

Parameter: (String) *sandbox*. The name of the sandbox to set as active.

Returns: `dojo.promise.Promise` as a promise that is resolved when the active sandbox is set.

### subset

Sets a subset object.

Parameter: (Object) *subset* An object that represents the dimension subset object to set. The object uses the following format:

#### sheetIndex

Type: Integer

The zero-based index of the sheet that contains the SUBNM cell whose dimension subset you want to change.

#### rowIndex

Type: Integer

The zero-based index of the row that contains the SUBNM cell whose dimension subset you want to change.

#### columnIndex

Type: Integer

The zero-based index of the column that contains the SUBNM cell whose dimension subset you want to change.

**dimension**

Type: String

The dimension name. Should not be used in conjunction with `sheetIndex`, `rowIndex`, and `columnIndex`.

**setExpression**

Type: String

The MDX expression used to define the subset. Not to be used in conjunction with `subset`. That is, either a `setExpression` or a subset name is provided from the input.

**subset**

Type: String

The subset name of the dimension subset to set. Not to be used in conjunction with `setExpression`.

**alias**

Type: String

The alias of the dimension subset to set.

**element**

Type: String

The name of the element. Not to be used with `elementIndex`.

**elementIndex**

Type: Integer

The one-based index of the dimension element to set. Not to be used with `element`.

Returns `dojo.Promise` as a promise that is resolved when the subset objects are set. Any callbacks that are registered with the promise are passed an object that matches the format of the subset that is passed into this method. A value of null is passed if the subset was not changed.

**subsets**

Sets multiple subset objects.

Parameter: (Object[]) *subsets* An array of subset objects to set. Each object uses following format:

**sheetIndex**

Type: Integer

The zero-based index of the sheet that contains the SUBNM cell whose dimension subset you want to change.

**rowIndex**

Type: Integer

The zero-based index of the row that contains the SUBNM cell whose dimension subset you want to change.

**columnIndex**

Type: Integer

The zero-based index of the column that contains the SUBNM cell whose dimension subset you want to change.

**dimension**

Type: String

The dimension name. Should not be used in conjunction with `sheetIndex`, `rowIndex`, and `columnIndex`.

**setExpression**

Type: String

The MDX expression used to define the subset. Not to be used in conjunction with `subset`. That is, either a `setExpression` or a subset is provided from the input.

**subset**

Type: String

The subset name of the dimension subset to set. Not to be used in conjunction with `setExpression`.

**alias**

Type: String

The alias of the dimension subset to set.

**element**

Type: String

The name of the element. Not to be used with `elementIndex`.

**elementIndex**

Type: Integer

The one-based index of the dimension element to set. Not to be used with `element`.

Returns `dojo.promise.Promise` as a promise that is resolved when the subset objects are set. Any callbacks that are registered with the promise are passed an array of objects that match the format of the subset objects that are passed into this method for the subsets that are successfully set.

**titleDimensionElement**

Sets a title dimension element.

Parameter: (Object) *element* An object that represents the title dimension elements to set. The object uses the following format:

**sheetIndex**

Type: Integer

The zero-based index of the sheet that contains the SUBNM cell whose dimension element you want to change.

**rowIndex**

Type: Integer

The zero-based index of the row that contains the SUBNM cell whose dimension element you want to change.

**columnIndex**

Type: Integer

The zero-based index of the column that contains the SUBNM cell whose dimension element you want to change.

**element**

Type: String

The name of the element. Not to be used with `elementIndex`.

**elementIndex**

Type: Integer

The one-based index of the dimension element to set. Not to be used with `element`.

Returns `dojo.promise.Promise` as a promise that is resolved when the title dimension element is set. Any callbacks that are registered with the promise are passed an object that matches the format of the element that is passed into this method. A value of null is passed if the element was not changed.

**titleDimensionElements**

Sets multiple title dimension elements.

Parameter: (Object[]) *elements* An array of the title dimension elements to set. Each object uses following format:

**sheetIndex**

Type: Integer

The zero-based index of the sheet that contains the SUBNM cell for the dimension element that you want to change. Optional when used with `dimension`, but required for `rowIndex` and `columnIndex`.

**rowIndex**

Type: Integer

The zero-based index of the row that contains the SUBNM cell for the dimension element that you want to change. Do not use this parameter with the dimension parameter.

**columnIndex**

Type: Integer

The zero-based index of the column that contains the SUBNM cell for the dimension element that you want to change. Do not use this parameter with the dimension parameter.

**dimension**

Type: String

The name of the dimension. Do not use this parameter with `rowIndex` and `columnIndex`.

**element**

Type: String

The name of the element. Not to be used with `elementIndex`.

**elementIndex**

Type: Integer

The one-based index of the dimension element to set. Not to be used with `element`.

Returns `dojo.promise.Promise` as a promise that is resolved when the title dimension elements are set. Any callbacks that are registered with the promise are passed an array of objects that match the format of the element objects that are passed into this method for the elements that are successfully set.

**Workbook methods**

The Workbook class has the following methods.

**startup**

Begins the startup sequence for this object. Call this function after the object is added to the document. The `onLoad` method is run after the startup sequence completes.

Applies to both CubeViewer and Workbook objects.

Syntax: `startup()`

Example:

```
document.body.appendChild(loadedWorksheet.domNode);
loadedWorksheet.startup();
```

See the Dojo documentation for [dijit.\\_WidgetBase#startup](#).

**commitActiveSandbox**

Commits changed data in the active sandbox to the base sandbox.

Returns `dojo.promise.Promise`. A promise that is resolved when the sandbox commit attempt completes. Any callbacks that are registered with the promise are passed a boolean with a value of true if the sandbox commit was successful or a value of false if the commit was not successful.

**copy**

Copies the selected cells to the clipboard if a selection exists.

**destroy**

Destroys this object and prepares it for garbage collection.

See Dojo documentation for [dijit.\\_WidgetBase#destroy](#).

**logout**

Destroys the TM1 Web session that is associated with this object's `sessionToken`.

Returns `dojo.promise.Promise` as a promise that is resolved when the logout completes.

**onActionButtonExecution**

Called when an action button is executed.

Syntax: `onActionButtonExecution: function(executionResults) {}`

Parameters: `executionResults` object that uses the following format.

**calculation**

Type: String

What type of calculation occurred on the current Workbook before action button execution occurred.

Value will be one of "None", "Recalculate", "Rebuild".

**navigation**

Type: Object

This property exists only if workbook or sheet navigation occurred as part of the action button execution.

**calculation**

Type: String

What type of calculation occurred on the target Workbook after the action button navigation occurred.

Value will be one of "None", "Recalculate", "Rebuild".

**objectId**

Type: String

The objectId of the workbook that has been navigated to. If an action on a worksheet within the same workbook occurred, the objectId will match the current workbook.

**path**

Type: String

The path to the workbook that was navigated to.

**name**

Type: String

The name of the target workbook.

**sheetIndex**

Type: Integer

The zero-based index of the worksheet that was navigated to.

**replace**

Type: Boolean

Whether the action button was configured to replace the existing workbook.

**tiProcess**

Type: Object

This property exists only if a TI process was executed as part of the action button's execution.

**calculation**

Type: String

What type of calculation occurred on the current workbook after the TI process was executed.

Value will be one of "None", "Recalculate", "Rebuild".

**name**

Type: String

The name of the TI process that was executed.

**executionSucceeded**

Type: Boolean

Whether or not the TI process execution was successful.

**onLoad**

Runs when the object is finished loading.

**onTitleDimensionElementChange**

Executed when a title dimension element is changed. Can be overridden during object construction or attached to using `dojo/aspect` module.

Syntax: `onTitleDimensionElementChange: function(elementInfo) {}`

Parameters: `elementInfo` object that uses the following format.

**sheetIndex**

Type: Integer

The zero-based index of the sheet that contains the SUBNM cell that was changed.

**rowIndex**

Type: Integer

The zero-based index of the row that contains the SUBNM cell that was changed.

**columnIndex**

Type: Integer

The zero-based index of the column that contains the SUBNM cell that was changed.

**dimension**

Type: String

The name of the dimension.

**element**

Type: String

The name of the element.

**elementIndex**

Type: Integer

The one-based index of the dimension element.

**paste**

Pastes the contents of the clipboard into the current selected area if a selection exists.

**rebuildActiveForms**

Rebuilds the active forms in the workbook.

Returns `dojo.promise.Promise` as a promise that is resolved when active forms are rebuilt.

**redo**

Performs a redo action.

Returns `dojo.promise.Promise` as a promise that is resolved when the redo action completes.

**replace**

Accepts an `objectId` and replaces the existing workbook with the one represented from the given `objectId` (unless it is the same as the existing worksheet, in which case nothing no action is taken).

Replace assumes that the workbook that is replacing the existing one uses the same TM1 Web session as the previous workbook.

**undo**

Performs an undo action.

Returns `dojo.promise.Promise` as a promise that is resolved when the undo action completes.

**TM1 Web JavaScript library CubeViewer class**

The `CubeViewer` class represents a TM1 Web `CubeViewer` object.

`CubeViewer` objects extend the Dojo widget object (`dijit._WidgetBase`) and can be assigned as a child object of a Dojo tab container (`dijit.layout.TabContainer`) or other container. For more information, see [Dojo documentation](http://dojotoolkit.org/documentation/) (<http://dojotoolkit.org/documentation/>).

In addition to the available properties and methods of the Dojo widget object, `CubeViewer` objects also have TM1 related properties and methods that you can access programmatically.

CubeViewer objects are loaded asynchronously and must finish loading before your code can interact with the objects.

## Format

You load a CubeViewer object by using the following format to specify the required properties and optional functions that define the object.

```
new CubeViewer({properties ..., functions ...});
```

## Properties

The *properties* include the following values that define the CubeViewer object.

- adminHost
- tm1Server
- username
- password
- camPassport
- sessionToken
- objectId
- view
- cube
- isPublic

**Note:** You can provide login credentials as either a session token and object ID, or by including separate values for TM1 Admin host, TM1 Server, user name, password, or camPassport.

## Functions

The *functions* can include the following optional code:

- Use the onLoad function so that you can be notified when the object is loaded and ready to interact with.
- Use the onTitleDimensionElementChange declaration so you can process the event when a user changes a dimension title in the related object.

## Example

The following example shows a JavaScript function that loads a CubeViewer object.

The login credentials are provided by using a session token.

```
function loadCubeview() {
  require([
    "tm1web/api/CubeViewer"
  ], function(CubeViewer) {
    var loadedCubeview = new CubeViewer{
      sessionToken: "yourSessionToken",
      cube: "plan_BudgetPlan",
      view: "Budget Input Detailed",
      isPublic: true,
      onLoad: function() {
        console.debug("CubeViewer loaded successfully.");
      },
    };
  });

  // Add cubeview to the document body
  document.body.appendChild(loadedCubeview.domNode);

  loadedCubeview.startup();
};
```

## CubeViewer properties

The CubeViewer class has the following properties.

When instantiating either a CubeViewer or Workbook object, the following properties are common between the two types of objects:

### sessionToken

Type: String

Specifies the TM1 Web session to use for this object. Do not use this property with the properties for `adminHost`, `tm1Server`, `username`, `password`, and `camPassport`. If this property is not specified, and no additional credentials are provided, the user is prompted with a login dialog during startup.

### objectId

Type: String

The ID of the CubeViewer. A unique number that you can use to reference the specific CubeViewer.

### adminHost

Type: String

Default: localhost

The admin host to use when the object is loaded. Do not use this property with the `sessionToken` property.

### tm1Server

Type: String

The TM1 server to use when the object is loaded. Do not use this property with the `sessionToken` property. If unspecified and no `sessionToken` is provided, the user is prompted with a login dialog during startup.

### username

Type: String

The user name to use when the object is loaded. Do not use this property with the `sessionToken` or `camPassport` properties. If unspecified and no `sessionToken` or `camPassport` is provided, the user is prompted with a login dialog during startup.

### password

Type: String

The password to use when the object is loaded. If unspecified and no `sessionToken` is provided, the user is prompted with a login dialog during startup.

### camPassport

Type: String

The Cognos BI authentication passport (CAM passport) to use when you load an object. Do not use this property with `username` or `sessionToken`.

### domNode

Type: HTML Element

The underlying HTML element that represents the widget. This property is automatically defined during object construction and should not be provided during instantiation.

For more information, see [Dojo documentation for dijit.\\_WidgetBase](https://dojotoolkit.org/reference-guide/1.10/dijit/_WidgetBase.html) ([https://dojotoolkit.org/reference-guide/1.10/dijit/\\_WidgetBase.html](https://dojotoolkit.org/reference-guide/1.10/dijit/_WidgetBase.html)).

The following properties are used when you instantiate a CubeViewer object only.

### view

Type: String

The name of the cube view to load.

### cube

Type: String

The name of the cube that contains the view you want to load.

## **isPublic**

Type: Boolean

Default: true

The access type of the cube view to load.

A value of true indicates that you want to load a public cube view.

A value of false indicates that you want to load a private cube view.

## **Get properties**

All properties that get a value are called with the following format:

```
get("property_Name").
```

For example: `get("sandboxes");`

### **sandboxes**

Retrieves all available sandboxes.

Returns `dojo.promise.Promise` as a promise that is resolved when the sandboxes are retrieved.

When the promise is resolved, an Array of objects that represent the available sandboxes is passed to any callback registered with the promise.

Each object uses the following format:

- name: (String) - The name of the sandbox.
- active: (Boolean) - True if this sandbox is the active sandbox for the object, else false.
- baseSandbox: (Boolean) - True if this sandbox is the base sandbox, else false.
- defaultSandbox: (Boolean) - True if this sandbox is the default sandbox, else false.

## **Set properties**

All properties that set a value are called with the following format:

```
set("property_Name", value)
```

For example: `set("activeSandbox", "theSandbox");`

### **activeSandbox**

Sets the specified sandbox as active.

Parameter: (String) *sandbox*. The name of the sandbox to set as active.

Returns: `dojo.promise.Promise` as a promise that is resolved when the active sandbox is set.

### **automaticRecalculation**

Sets automatic recalculation to on or off.

Parameters: Boolean.

- True turns on automatic recalculation.
- False turns off automatic recalculation.

Returns: `dojo.promise.Promise`. A promise that is resolved when automatic recalculation is enabled or disabled.

### **chartType**

Sets the chart type of the CubeViewer object.

Parameters: `tm1web.cubeview.ChartType`. The chart type to set.

Returns: `dojo.promise.Promise`. A promise that is resolved when the chart type is set.

### **dimensionBarVisible**

Sets the visibility of the dimension bar.

Parameters: Boolean.

- `True` turns on the display of the dimension bar.
- `False` turns off the display of the dimension bar.

### **displayMode**

Sets the display mode of the CubeViewer object.

Parameters: `tm1web.cubeview.DisplayMode`. The display mode to set.

Returns: `dojo.promise.Promise`. A promise that is resolved when the display mode is set.

### **subset**

Sets a subset object.

Parameter: (Object) *subset* An object that represents the dimension subset object to set. The object uses the following format:

#### **dimension**

Type: String

The dimension name.

#### **setExpression**

Type: String

The MDX expression used to define the subset. Not to be used in conjunction with `subset`. That is, either a `setExpression` or a subset name is provided from the input.

#### **subset**

Type: String

The subset name of the dimension subset to set. Not to be used in conjunction with `setExpression`.

#### **alias**

Type: String

The alias of the dimension subset to set.

#### **element**

Type: String

The name of the element. Not to be used with `elementIndex`.

#### **elementIndex**

Type: Integer

The one-based index of the dimension element to set. Not to be used with `element`.

Returns `dojo.promise.Promise` as a promise that is resolved when the subset objects are set. Any callbacks that are registered with the promise are passed an object that matches the format of the subset that is passed into this method. A value of `null` is passed if the subset was not changed.

### **subsets**

Sets multiple subset objects.

Parameter: (Object[]) *subsets* An array of subset objects to set. Each object uses following format:

#### **dimension**

Type: String

The dimension name.

#### **setExpression**

Type: String

The MDX expression used to define the subset. Not to be used in conjunction with `subset`. That is, either a `setExpression` or a subset is provided from the input.

#### **subset**

Type: String

The subset name of the dimension subset to set. Not to be used in conjunction with `setExpression`.

**alias**

Type: String

The alias of the dimension subset to set.

**element**

Type: String

The name of the element. Not to be used with `elementIndex`.

**elementIndex**

Type: Integer

The one-based index of the dimension element to set. Not to be used with `element`.

Returns `dojo.promise.Promise` as a promise that is resolved when the subset objects are set. Any callbacks that are registered with the promise are passed an array of objects that match the format of the subset objects that are passed into this method for the subsets that are successfully set.

**titleDimensionElement**

Sets a title dimension element.

Parameter: element object. The title dimension element to set. This object uses the following format:

**dimension**

String

The name of the dimension.

**element**

String

The name of the element. Do not use this parameter with `elementIndex`.

**elementIndex**

Integer

The one-based index of the dimension element to set. Do not use this parameter with the `element` parameter.

Returns: `dojo.promise.Promise`. A promise that is resolved when the title dimension element is set. Any callbacks that are registered with the promise are passed an object that matches the format of the element that was passed into this method. A value of null is passed if the element was not changed.

**titleDimensionElements**

Sets multiple title dimension elements.

Parameter: `object[]` elements. An array of the title dimension elements to set. Each object uses the following format:

**dimension**

String

The name of the dimension.

**element**

String

The name of the element. Do not use this parameter with `elementIndex`.

**elementIndex**

Integer

The one-based index of the dimension element to set. Do not use this parameter with the `element` parameter.

Returns `dojo.promise.Promise`. A promise that is resolved when the title dimension elements are set. Any callbacks that are registered with the promise are passed an array of objects that match the format of the element objects that were passed into this method. The passed array reports back about the elements that are successfully set.

## CubeViewer methods

The CubeViewer class has the following methods.

### startup

Begins the startup sequence for this object. Call this function after the object is added to the document. The onLoad method is run after the startup sequence completes.

Applies to both CubeViewer and Workbook objects.

Syntax: startup()

Example:

```
document.body.appendChild(loadedCubeViewer.domNode);
loadedCubeViewer.startup();
```

See the Dojo documentation for [dijit.\\_WidgetBase#startup](#).

### commitActiveSandbox

Commits changed data in the active sandbox to the base sandbox.

Returns `dojo.promise.Promise`. A promise that is resolved when the sandbox commit attempt is completed. Any callbacks that are registered with the promise are passed a boolean with a value of true if the sandbox commit was successful. A value of false is passed if the commit was unsuccessful.

### copy

Copies the selected cells to the clipboard if a selection exists.

### destroy

Destroys this object and prepares it for garbage collection.

See Dojo documentation for [dijit.\\_WidgetBase#destroy](#).

### logout

Destroys the TM1 Web session that is associated with this object's sessionToken.

Returns `dojo.promise.Promise` as a promise that is resolved when the logout completes.

### onLoad

Runs when the object is finished loading.

### onTitleDimensionElementChange

Executed when a title dimension element is changed. Can be overridden during object construction or attached to by using the `dojo/aspect` module.

Syntax: `onTitleDimensionElementChange: function(elementInfo) {}`

Parameter: `elementInfo` object. This object uses the following format:

#### dimension

String

The name of the dimension that was changed.

#### element

String

The name of the element that was changed.

#### elementIndex

Integer

The one-based index of the dimension element that was changed.

### paste

Pastes the contents of the clipboard into the current selected area if a selection exists.

### redo

Performs a redo action.

Returns `dojo.promise.Promise` as a promise that is resolved when the redo action completes.

**reset**

Resets the cube view to its original saved state.

Returns: `dojo.Promise`. A promise that is resolved when the cube view is reset.

**save**

Saves the layout of cube view and overwrites the existing layout.

Returns: `dojo.Promise`. A promise that is resolved when the cube view is saved.

**undo**

Performs an undo action.

Returns `dojo.Promise` as a promise that is resolved when the undo action completes.



---

## Appendix B. Supported Microsoft Excel functions - TM1 Web

IBM TM1 Web supports many Excel worksheet functions. This appendix lists the supported Excel functions by category and in alphabetical order, and describes any differences in performance between the Excel functions and TM1 Web functions.

### Date and Time functions

---

The following table lists date and time functions.

Function	Description
DATE	Returns the serial number of a particular date.
DATEVALUE	Converts a date in the form of text to a serial number.
DAY	Converts a serial number to a day of the month.
DAYS360	Calculates the number of days between two dates based on a 360-day year.
HOUR	Converts a serial number to an hour.
MINUTE	Converts a serial number to a minute.
MONTH	Converts a serial number to a month.
NOW	Returns the serial number of the current date and time.
SECOND	Converts a serial number to a second.
TIME	Returns the serial number of a particular time.
TIMEVALUE	Converts a time in the form of text to a serial number.
TODAY	Returns the serial number of today's date.
WEEKDAY	Converts a serial number to a day of the week.
YEAR	Converts a serial number to a year.

### Financial functions

---

The following table lists financial functions.

Function	Description
DB	Returns the depreciation of an asset for a specified period using the fixed-declining balance method.

<b>Function</b>	<b>Description</b>
DDB	Returns the depreciation of an asset for a specified period using the double-declining balance method or some other method you specify.
FV	Returns the future value of an investment.
IPMT	Returns the interest payment for an investment for a given period.
IRR	Returns the internal rate of return for a series of cash flows.
ISPMT	Calculates the interest paid during a specific period of an investment.
MIRR	Returns the internal rate of return where positive and negative cash flows are financed at different rates.
NPER	Returns the number of periods for an investment.
NPV	Returns the net present value of an investment based on a series of periodic cash flows and a discount rate.
PMT	Returns the periodic payment for an annuity.
PPMT	Returns the payment on the principal for an investment for a given period.
PV	Returns the present value of an investment.
RATE	Returns the interest rate per period of an annuity.
SLN	Returns the straight-line depreciation of an asset for one period.
SYD	Returns the sum-of-years' digits depreciation of an asset for a specified period.

## Information functions

The following table lists information functions that are supported in TM1 Web.

<b>Function</b>	<b>Description</b>
CELL	Returns information about the formatting, location, or contents of a cell. Support for the Cell function is limited to the following info_types: address, col, row, protect, contents, type.
ISERR	Returns TRUE if the value is any error value except #N/A.
ISERROR	Returns TRUE if the value is any error value.
ISNA	Returns TRUE if the value is the #N/A error value.
NA	Returns the error value #N/A.

## Logical functions

---

The following table lists logical functions.

Function	Description
AND	Returns TRUE if all its arguments are TRUE.
FALSE	Returns the logical value FALSE.
IF	Specifies a logical test to perform.
NOT	Reverses the logic of its argument.
OR	Returns TRUE if any argument is TRUE.
TRUE	Returns the logical value TRUE.

## Lookup and reference functions

---

The following table lists lookup and reference functions.

**Note:** Certain functions, such as LOOKUP and ROWS, may accept two dimensional arrays as arguments. TM1 Web does not support two dimensional arrays. Depending on the data organization and requirements, these functions can still obtain correct values, for example, when the data being retrieved falls in the initial portions of the array. To ensure correct values when working with these functions on TM1 Web you may need to reorganize the input data into repeated functions using one dimensional arrays or you may need to use direct cell references.

Function	Description
ADDRESS	Returns a reference as text to a single cell in a worksheet.
CHOOSE	Chooses a value from a list of values.
COLUMN	Returns the column number of a reference.
COLUMNS	Returns the number of columns in a reference.
HLOOKUP	Looks in the top row of an array and returns the value of the indicated cell.
HYPERLINK	Creates a shortcut or jump that opens a document stored on a network server, an intranet, or the Internet.
INDEX	Uses an index to choose a value from a reference or array.
LOOKUP	Looks up values in a vector or array.
MATCH	Looks up values in a reference or array.
OFFSET	Returns a reference offset from a given reference.
ROW	Returns the row number of a reference.

Function	Description
ROWS	Returns the number of rows in a reference.
VLOOKUP	Looks in the first column of an array and moves across the row to return the value of a cell.

## Math and trigonometric functions

The following table lists math and trigonometric functions.

Function	Description
ABS	Returns the absolute value of a number.
ACOS	Returns the arccosine of a number.
ACOSH	Returns the inverse hyperbolic cosine of a number.
ASIN	Returns the arcsine of a number.
ASINH	Returns the inverse hyperbolic sine of a number.
ATAN	Returns the arctangent of a number.
ATAN2	Returns the arctangent from x- and y-coordinates.
ATANH	Returns the inverse hyperbolic tangent of a number.
CEILING	Rounds a number to the nearest integer or to the nearest multiple of significance.
COMBIN	Returns the number of combinations for a given number of objects.
COS	Returns the cosine of a number.
COSH	Returns the hyperbolic cosine of a number.
DEGREES	Converts radians to degrees.
EVEN	Rounds a number up to the nearest even integer.
EXP	Returns e raised to the power of a given number.
FACT	Returns the factorial of a number.
FLOOR	Rounds a number down, toward zero.
INT	Rounds a number down to the nearest integer.
LN	Returns the natural logarithm of a number.
LOG	Returns the logarithm of a number to a specified base.

Function	Description
LOG10	Returns the base-10 logarithm of a number.
MOD	Returns the remainder from division.
ODD	Rounds a number up to the nearest odd integer.
PI	Returns the value of pi.
POWER	Returns the result of a number raised to a power.
PRODUCT	Multiplies its arguments.
RADIANS	Converts degrees to radians.
RAND	Returns a random number between 0 and 1.
ROMAN	Converts an Arabic numeral to a Roman numeral, as text.
ROUND	Rounds a number to a specified number of digits.
ROUNDDOWN	Rounds a number down, toward zero.
ROUNDUP	Rounds a number up, away from zero.
SIGN	Returns the sign of a number.
SIN	Returns the sine of the given angle.
SINH	Returns the hyperbolic sine of a number.
SQRT	Returns a positive square root.
SUM	Adds its arguments.
SUMIF	<p>Adds the values in a range specified by a given criteria.</p> <p><b>Note:</b> You must use a fixed size range in the criteria to sum. Using a full column reference is not supported. For example, to represent the following formula:</p> <p>=SUMIF(Calculation!\$B:\$B,\$B21,Calculation!\$M:\$M)</p> <p>You must use:</p> <p>=SUMIF(Calculation!\$B\$1:\$B\$500,\$B21,Calculation!\$M\$1:\$M\$500)</p> <p>If you specify a sum_range as an optional parameter, it must be the same size as the range parameter.</p>
TAN	Returns the tangent of a number.
TANH	Returns the hyperbolic tangent of a number.

## Text and data functions

---

The following table lists text and data functions.

<b>Function</b>	<b>Description</b>
CHAR	Returns the character specified by the code number.
CLEAN	Removes all nonprintable characters from text.
CODE	Returns a numeric code for the first character in a text string.
CONCATENATE	Joins several text items into one text item.
DOLLAR	Converts a number to text, using the \$ (dollar) currency format.
EXACT	Checks to see if two text values are identical.
FIND	Finds one text value within another (case-sensitive).
FIXED	Formats a number as text with a fixed number of decimals.
LEFT	Returns the leftmost characters from a text value.
LEN	Returns the number of characters in a text string.
LOWER	Converts text to lowercase.
MID	Returns a specific number of characters from a text string starting at the position you specify.
PROPER	Capitalizes the first letter in each word of a text value.
REPLACE	Replaces characters within text.
REPT	Repeats text a given number of times.
RIGHT	Returns the rightmost characters from a text value.
SEARCH	Finds one text value within another (not case-sensitive).
SUBSTITUTE	Substitutes new text for old text in a text string.
T	Converts its arguments to text.
TEXT	Formats a number and converts it to text.
TRIM	Removes spaces from text.
UPPER	Converts text to uppercase.
VALUE	Converts a text argument to a number.

## Statistical functions

---

The following table lists statistical functions.

<b>Function</b>	<b>Description</b>
AVEDEV	Returns the average of the absolute deviations of data points from their mean.
AVERAGE	Returns the average of its arguments.
AVERAGEA	Returns the average of its arguments, including numbers, text, and logical values.
BINOMDIST	Returns the individual term binomial distribution probability.
CONFIDENCE	Returns the confidence interval for a population mean.
CORREL	Returns the correlation coefficient between two data sets.
COUNT	Counts how many numbers are in the list of arguments.
COUNTA	Counts how many values are in the list of arguments.
COUNTIF	Counts the number of nonblank cells within a range that meet the given criteria.
COVAR	Returns covariance, the average of the products of paired deviations.
DEVSQ	Returns the sum of squares of deviations.
EXPONDIST	Returns the exponential distribution.
FISHER	Returns the Fisher transformation.
FISHERINV	Returns the inverse of the Fisher transformation.
FORECAST	Returns a value along a linear trend.
GEOMEAN	Returns the geometric mean.
GROWTH	Returns values along an exponential trend.
HARMEAN	Returns the harmonic mean.
INTERCEPT	Returns the intercept of the linear regression line.
KURT	Returns the kurtosis of a data set.
LARGE	Returns the k-th largest value in a data set.
LINEST	Returns the parameters of a linear trend.
LOGEST	Returns the parameters of an exponential trend.

<b>Function</b>	<b>Description</b>
MAX	Returns the maximum value in a list of arguments.
MATCH	Returns the relative position of an item in an array that matches a specified value in a specified order.
MAXA	Returns the maximum value in a list of arguments, including numbers, text, and logical values.
MEDIAN	Returns the median of the given numbers.
MIN	Returns the minimum value in a list of arguments.
MINA	Returns the smallest value in a list of arguments, including numbers, text, and logical values.
NEGBINOMDIST	Returns the negative binomial distribution, the probability that there will be Number_f failures before the Number_s-th success, with Probability_f probability of a success.
MODE	Returns the most common value in a data set.
NORMDIST	Returns the normal cumulative distribution.
NORMINV	Returns the inverse of the normal cumulative distribution.
NORMSDIST	Returns the standard normal cumulative distribution.
NORMSINV	Returns the inverse of the standard normal cumulative distribution.
PEARSON	Returns the Pearson product moment correlation coefficient.
PERMUT	Returns the number of permutations for a given number of objects.
RSQ	Returns the square of the Pearson product moment correlation coefficient.
SKEW	Returns the skewness of a distribution.
SLOPE	Returns the slope of the linear regression line.
SMALL	Returns the k-th smallest value in a data set.
STANDARDIZE	Returns a normalized value.
STDEV	Estimates standard deviation based on a sample.
STDEVA	Estimates standard deviation based on a sample, including numbers, text, and logical values.
STDEVP	Calculates standard deviation based on the entire population.
STDEVPA	Calculates standard deviation based on the entire population, including numbers, text, and logical values.

<b>Function</b>	<b>Description</b>
STEYX	Returns the standard error of the predicted y-value for each x in the regression.
TREND	Returns values along a linear trend.
VAR	Estimates variance based on a sample.
VARA	Estimates variance based on a sample, including numbers, text, and logical values.
VARP	Calculates variance based on the entire population.
VARPA	Calculates variance based on the entire population, including numbers, text, and logical values.
WEIBULL	Returns the Weibull distribution.



---

## Appendix C. Unsupported Microsoft Excel functions - TM1 Web

IBM TM1 Web supports many Excel worksheet functions. This appendix lists the Excel functions, by category and in alphabetical order, that are not supported in TM1 Web.

### Database and list management functions

---

This table lists the management functions that are not supported in TM1 Web.

Function	Description
DAVERAGE	Returns the average of selected database entries.
DCOUNT	Counts the cells that contain numbers in a database.
DCOUNTA	Counts nonblank cells in a database.
DGET	Extracts from a database a single record that matches the specified criteria.
DMAX	Returns the maximum value from selected database entries.
DMIN	Returns the minimum value from selected database entries.
DPRODUCT	Multiplies the values in a particular field of records that match the criteria in a database.
DSTDEV	Estimates the standard deviation based on a sample of selected database entries.
DSTDEVP	Calculates the standard deviation based on the entire population of selected database entries.
DSUM	Adds the numbers in the field column of records in the database that match the criteria.
DVAR	Estimates variance based on a sample from selected database entries.
DVARP	Calculates variance based on the entire population of selected database entries.

### Date and time functions

---

This table lists the date and time functions that are not supported in TM1 Web.

Function	Description
EDATE	Returns the serial number of the date that is the indicated number of months before or after the start date.

Function	Description
EOMONTH	Returns the serial number of the last day of the month before or after a specified number of months.
NETWORKDAYS	Returns the number of whole workdays between two dates.
WEEKNUM	Converts a serial number to a number representing where the week falls numerically with a year.
WORKDAY	Returns the serial number of the date before or after a specified number of workdays.
YEARFRAC	Returns the year fraction representing the number of whole days between start_date and end_date.

## Financial functions

This table lists the financial functions that are not supported in TM1 Web.

Functions	Description
ACCRINT	Returns the accrued interest for a security that pays periodic interest.
ACCRINTM	Returns the accrued interest for a security that pays interest at maturity.
AMORDEGRC	Returns the depreciation for each accounting period by using a depreciation coefficient.
AMORLINC	Returns the depreciation for each accounting period.
COUPDAYBS	Returns the number of days from the beginning of the coupon period to the settlement date.
COUPDAYS	Returns the number of days in the coupon period that contains the settlement date.
COUPDAYSNC	Returns the number of days from the settlement date to the next coupon date.
COUPNCD	Returns the next coupon date after the settlement date.
COUPNUM	Returns the number of coupons payable between the settlement date and maturity date.
COUPPCD	Returns the previous coupon date before the settlement date.
CUMIPMT	Returns the cumulative interest paid between two periods.
CUMPRINC	Returns the cumulative principal paid on a loan between two periods.
DISC	Returns the discount rate for a security.

<b>Functions</b>	<b>Description</b>
DOLLARDE	Converts a dollar price, expressed as a fraction, into a dollar price, expressed as a decimal number.
DOLLARFR	Converts a dollar price, expressed as a decimal number, into a dollar price, expressed as a fraction.
DURATION	Returns the annual duration of a security with periodic interest payments.
EFFECT	Returns the effective annual interest rate.
FVSCHEDULE	Returns the future value of an initial principal after applying a series of compound interest rates.
INTRATE	Returns the interest rate for a fully invested security.
MDURATION	Returns the Macauley modified duration for a security with an assumed par value of \$100.
NOMINAL	Returns the annual nominal interest rate.
ODDFPRICE	Returns the price per \$100 face value of a security with an odd first period.
ODDFYIELD	Returns the yield of a security with an odd first period.
ODDLPRICE	Returns the price per \$100 face value of a security with an odd last period.
ODDLYIELD	Returns the yield of a security with an odd last period.
PRICE	Returns the price per \$100 face value of a security that pays periodic interest.
PRICEDISC	Returns the price per \$100 face value of a discounted security.
PRICEMAT	Returns the price per \$100 face value of a security that pays interest at maturity.
RECEIVED	Returns the amount received at maturity for a fully invested security.
TBILLEQ	Returns the bond-equivalent yield for a Treasury bill.
TBILLPRICE	Returns the price per \$100 face value for a Treasury bill.
TBILLYIELD	Returns the yield for a Treasury bill.
VDB	Returns the depreciation of an asset for a specified or partial period using a declining balance method.
XIRR	Returns the internal rate of return for a schedule of cash flows that is not necessarily periodic.
XNPV	Returns the net present value for a schedule of cash flows that is not necessarily periodic.

<b>Functions</b>	<b>Description</b>
YIELD	Returns the yield on a security that pays periodic interest.
YIELDDISC	Returns the annual yield for a discounted security; for example, a Treasury bill.
YIELDMAT	Returns the annual yield of a security that pays interest at maturity.

## Information functions

This table lists the information functions that are not supported in TM1 Web.

<b>Function</b>	<b>Description</b>
ERROR.TYPE	Returns a number corresponding to an error type.
IFERROR	Returns a value you specify if a formula evaluates to an error.
INFO	Returns information about the current operating environment.
ISBLANK	Returns TRUE if the value is blank.
ISEVEN	Returns TRUE if the number is even.
ISLOGICAL	Returns TRUE if the value is a logical value.
ISNONTEXT	Returns TRUE if the value is not text.
ISNUMBER	Returns TRUE if the value is a number.
ISODD	Returns TRUE if the number is odd.
ISREF	Returns TRUE if the value is a reference.
ISTEXT	Returns TRUE if the value is text.
N	Returns a value converted to a number.
TYPE	Returns a number indicating the data type of a value.

## Lookup and reference functions

This table lists the lookup and reference functions that are not supported in TM1 Web.

<b>Function</b>	<b>Description</b>
AREAS	Returns the number of areas in a reference.
INDIRECT	Returns a reference indicated by a text value.
RTD	Retrieves real-time data from a program that supports COM automation.

Function	Description
TRANSPOSE	Returns the transpose of an array.

## Math and trigonometric functions

This table lists the math and trigonometric functions that are not supported in TM1 Web.

Function	Description
FACTDOUBLE	Returns the double factorial of a number.
GCD	Returns the greatest common divisor.
LCM	Returns the least common multiple.
MDETERM	Returns the matrix determinant of an array.
MINVERSE	Returns the matrix inverse of an array.
MMULT	Returns the matrix product of two arrays.
MROUND	Returns a number rounded to the desired multiple.
MULTINOMIAL	Returns the multinomial of a set of numbers.
QUOTIENT	Returns the integer portion of a division.
RANDBETWEEN	Returns a random number between the numbers you specify.
SERIESSUM	Returns the sum of a power series based on the formula.
SQRTPI	Returns the square root of (number * pi).
SUBTOTAL	Returns a subtotal in a list or database.
SUMIFS	Adds all of the arguments that meet multiple criteria.
SUMPRODUCT	Returns the sum of the products of corresponding array components.
SUMSQ	Returns the sum of the squares of the arguments.
SUMX2MY2	Returns the sum of the difference of squares of corresponding values in two arrays.
SUMX2PY2	Returns the sum of the sum of squares of corresponding values in two arrays.
SUMXMY2	Returns the sum of squares of differences of corresponding values in two arrays.
TRUNC	Truncates a number to an integer.

## Statistical functions

This table lists the statistical functions that are not supported in TM1 Web.

Function	Description
BETADIST	Returns the beta cumulative distribution function.
BETAINV	Returns the inverse of the cumulative distribution function for a specified beta distribution.
CHIDIST	Returns the one-tailed probability of the chi-squared distribution.
CHIINV	Returns the inverse of the one-tailed probability of the chi-squared distribution.
CHITEST	Returns the test for independence.
COUNTBLANK	Counts the number of blank cells within a range.
COUNTIFS	Counts the number of non-blank cells across multiple ranges that meet the given criteria.
CRITBINOM	Returns the smallest value for which the cumulative binomial distribution is less than or equal to a criterion value.
FDIST	Returns the F probability distribution.
FINV	Returns the inverse of the F probability distribution.
FREQUENCY	Returns a frequency distribution as a vertical array.
FTEST	Returns the result of an F-test.
GAMMADIST	Returns the gamma distribution.
GAMMAINV	Returns the inverse of the gamma cumulative distribution.
GAMMALN	Returns the natural logarithm of the gamma function, $\Gamma(x)$ .
HYPGEOMDIST	Returns the hyper geometric distribution.
LOGINV	Returns the inverse of the lognormal distribution.
LOGNORMDIST	Returns the cumulative lognormal distribution.
NEGBINOMDIST	Returns the negative binomial distribution.
PERCENTILE	Returns the k-th percentile of values in a range.
PERCENTRANK	Returns the percentage rank of a value in a data set.
POISSON	Returns the Poisson distribution.

<b>Function</b>	<b>Description</b>
PROB	Returns the probability that values in a range are between two limits.
QUARTILE	Returns the quartile of a data set.
RANK	Returns the rank of a number in a list of numbers.
TDIST	Returns the Student's t-distribution.
TINV	Returns the inverse of the Student's t-distribution.
TRIMMEAN	Returns the mean of the interior of a data set.
TTEST	Returns the probability associated with a Student's t-test.
ZTEST	Returns the one-tailed probability-value of a z-test.

## Text and data functions

---

This table lists the text and data functions that are not supported in TM1 Web.

<b>Function</b>	<b>Description</b>
ASC	Changes full-width (double-byte) English letters or katakana within a character string to half-width (single-byte) characters.
BAHTTEXT	Converts a number to text, using the ฿ (baht) currency format.
JIS	Changes half-width (single-byte) English letters or katakana within a character string to full-width (double-byte) characters.
PHONETIC	Extracts the phonetic (furigana) characters from a text string.
AutoShapes	TM1 Web does not support Microsoft Office Autoshapes.



## Notices

---

This information was developed for products and services offered worldwide.

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

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

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

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

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

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

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

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

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

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

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

IBM Software Group  
Attention: Licensing

3755 Riverside Dr.  
Ottawa, ON  
K1V 1B7  
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

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

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

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

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

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

#### COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_.

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

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



## Product Information

This document applies to IBM Planning Analytics 2.0.0 and may also apply to subsequent releases.

## Copyright

Licensed Materials - Property of IBM

© Copyright IBM Corp. 2007, 2018.

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

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web in "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

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

- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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





---

# Index

## Special Characters

.xlsx worksheet [89](#)

## A

Add command [32](#)

Admin Host [20](#)

administration

tm1web\_config.xml [70](#)

AIX [90](#)

API

JavaScript library [118](#)

session token login [91](#)

URL API [97](#)

## C

change password [69](#)

charts

drill through [39](#)

ClearType Tuner utility

on end user computers [11](#)

collapsing consolidations [29](#), [48](#)

column width in websheets [11](#)

columns

hide [10](#)

conditional formatting [10](#)

configuration parameters

TM1 Web [71](#)

configure login page [77](#)

consolidated cells in cube viewer [35](#)

consolidations

collapsing in a subset [48](#)

expand [47](#)

moving in a subset [44](#)

converting .xls [89](#)

converting worksheets [89](#)

creating views [36](#)

Cube Viewer

chart drill through [39](#)

collapsing consolidations [29](#)

drill assignments [31](#)

drill processes [31](#)

drilling [29](#)

editing data in cells [31](#)

expanding consolidations [29](#)

filtering [30](#)

generating reports [21](#), [37](#)

moving dimensions [29](#)

navigation [27](#)

opening [25](#)

page size [85](#)

paging toolbar [27](#)

pivoting dimensions [29](#)

recalculating data [28](#)

rolling up [29](#)

Cube Viewer (*continued*)

saving data [28](#)

spreading data [32](#)

stacking dimension [29](#)

Subset Editor [30](#)

toolbar [26](#)

write back [31](#)

Cube Viewer, export sheets [85](#)

CubeViewer class

methods [142](#)

properties [138](#)

CubeViewer objects

with JavaScript library [123](#)

with URL API [107](#)

CubeviewerStringWrap [85](#)

custom consolidations

from existing subsets [49](#)

from selected elements [50](#)

Custom scorecard diagrams

overview [65](#)

viewing [68](#)

## D

data

spreading [32](#)

data entry commands [32](#), [33](#)

data spreading

excluding [15](#), [35](#)

in a Cube View [32](#)

in a worksheet [14](#)

DEBUG

TM1 Web message severity [87](#)

Decrease command [32](#)

deleting

elements [44](#)

dimension

list [20](#)

pivoting [29](#)

stacking [29](#)

Divide command [32](#)

drilling

assignments [31](#)

processes [31](#)

dynamic subsets [41](#)

## E

edit

data in a worksheet [13](#)

enabling/disabling data in a worksheet [20](#)

subsets [41](#)

editing tm1web\_config.xml [70](#)

elements

deleting [44](#)

filtering [45](#)

inserting parents [49](#)

elements (*continued*)  
    keeping [44](#)  
    reducing in a subset [44](#)  
    sorting [47](#)  
entering data  
    data entry commands [32](#), [33](#)  
ERROR  
    TM1 Web message severity [87](#)  
Excel  
    worksheet functions [145](#)  
    worksheet functions unsupported [155](#)  
expand  
    consolidations [29](#), [47](#)  
export  
    maximum sheets [85](#)  
    reports [37](#)

## F

filtering  
    by attribute [45](#)  
    by expression [46](#)  
    by level [45](#)  
    data in Cube Viewer [30](#)  
    elements [45](#)  
    types [30](#)  
fonts  
    Microsoft Excel default [90](#)  
freezing  
    panes [11](#)  
functions  
    date and time [145](#)  
    financial [145](#)  
    information [146](#)  
    logical [147](#)  
    lookup and reference [147](#)  
    math and trigonometric [148](#)  
    statistical [151](#)  
    supported Excel worksheet [145](#)  
    text and data [150](#)  
    unsupported [155](#), [156](#), [158–161](#)

## G

generating reports [21](#), [37](#)  
Grow commands [32](#)

## H

hide columns [10](#)  
Hold command [32](#)  
hyperlinks [10](#)

## I

Impact diagram  
    overview [63](#)  
    viewing [67](#)  
Increase command [32](#)  
INFO  
    TM1 Web message severity [87](#)  
inserting  
    parents [49](#)

## J

JavaScript library  
    callback functions [124](#)  
    CubeViewer class [136](#)  
    CubeViewer methods [142](#)  
    CubeViewer properties [138](#)  
    HTML head tags [119](#)  
    loading CubeViewer objects [123](#)  
    loading worksheet objects [122](#)  
    overview [118](#)  
    property and method samples [126](#)  
    session token login [91](#)  
    Workbook class [128](#)  
    Workbook methods [134](#)  
    Workbook properties [129](#)  
Job Queuing [57](#)  
JobQueuing\_window [57](#)

## K

K command [32](#)  
keeping elements [44](#)

## L

LegacyUrlApiSessionDiscoveryEnabled parameter [93](#)  
login page configuring [77](#)

## M

M command [32](#)  
Metrics cube  
    overview [62](#)  
    viewing [66](#)  
Microsoft Excel  
    default font [90](#)  
moving dimensions [29](#)  
Multiply command [32](#)

## N

navigation  
    Cube Viewer [27](#)  
navigation tree  
    views node [84](#)

## P

Pagination [27](#)  
paging toolbar [27](#)  
parameters, tm1web\_config.xml [70](#)  
passwords  
    change [69](#)  
PDF  
    reports [37](#)  
Percent command [32](#)  
pivoting dimensions [29](#)  
Power command [32](#)  
print properties [20](#)  
publishing worksheets  
    defined [9](#)

## Q

- quick commands
  - data entry commands [32](#), [33](#)

## R

- recalculating data in Cube Viewer [28](#)
- reports
  - Cube Viewer [21](#), [37](#)
  - exporting [37](#)
  - overview [21](#), [37](#)
  - PDF [21](#), [37](#)
  - slice [21](#), [37](#)
  - snapshot [21](#), [37](#)
  - websheets [21](#), [37](#)
- rolling up [29](#)

## S

- sandbox
  - cell coloring [56](#)
  - committing [56](#)
  - deleting [51](#)
  - overview [51](#)
  - resetting data values [55](#)
- saving data in Cube Viewer [28](#)
- Scorecarding
  - Custom diagrams [68](#)
  - Impact diagram [63](#), [67](#)
  - Metrics cube [66](#)
  - overview [61](#)
  - Strategy map [67](#)
- selecting elements [30](#)
- session token login [91](#)
- shortcuts [33](#)
- slices
  - export reports [21](#)
  - reports [37](#)
- snapshot
  - reports [37](#)
- sorting elements [47](#)
- spreading
  - data [32](#)
- spreading data
  - excluding cells in a Cube View [35](#)
  - excluding cells in a worksheet [15](#)
  - excluding consolidations in a Cube View [35](#)
  - excluding consolidations in a worksheet [15](#)
  - in a Cube View [32](#)
  - in a worksheet [14](#)
- stacking dimensions [29](#)
- static
  - subsets [41](#)
- Strategy map
  - overview [64](#)
  - viewing [67](#)
- String measurement [11](#)
- Subset Editor
  - accessing [41](#)
  - collapse consolidations [48](#)
  - collapse tree fully [48](#)
  - displaying [41](#)

- Subset Editor (*continued*)
  - drill-down consolidations [48](#)
  - expand consolidations [47](#)
  - expand tree fully [48](#)
  - toolbar [42](#)
- subsets
  - collapsing consolidations [48](#)
  - deleting subsets [44](#)
  - dynamic [41](#)
  - editing [41](#)
  - expanding consolidations [47](#)
  - filtering elements [45](#)
  - inserting parents [49](#)
  - keeping elements [44](#)
  - moving consolidations [44](#)
  - moving elements [43](#)
  - selecting elements [30](#)
  - sorting elements [47](#)
- Subtract command [32](#)
- supported Excel functions
  - ABS [148](#)
  - ACOS [148](#)
  - ACOSH [148](#)
  - ADDRESS [147](#)
  - AND [147](#)
  - ASIN [148](#)
  - ASINH [148](#)
  - ATAN [148](#)
  - ATAN2 [148](#)
  - ATANH [148](#)
  - AVEDEV [151](#)
  - AVERAGE [151](#)
  - AVERAGEA [151](#)
  - BINOMDIST [151](#)
  - CEILING [148](#)
  - CELL [146](#)
  - CHAR [150](#)
  - CHOOSE [147](#)
  - CLEAN [150](#)
  - CODE [150](#)
  - COLUMN [147](#)
  - COLUMNS [147](#)
  - COMBIN [148](#)
  - CONCATENATE [150](#)
  - CONFIDENCE [151](#)
  - CORREL [151](#)
  - COS [148](#)
  - COSH [148](#)
  - COUNT [151](#)
  - COUNTA [151](#)
  - COUNTIF [151](#)
  - COVAR [151](#)
  - DATE [145](#)
  - DATEVALUE [145](#)
  - DAY [145](#)
  - DAYS360 [145](#)
  - DB [145](#)
  - DDB [146](#)
  - DEGREE [148](#)
  - DEVSQ [151](#)
  - DOLLAR [150](#)
  - EVEN [148](#)
  - EXACT [150](#)
  - EXP [148](#)

supported Excel functions (*continued*)

[EXPONDIST 151](#)  
[FACT 148](#)  
[FALSE 147](#)  
[FIND 150](#)  
[FISHER 151](#)  
[FISHERINV 151](#)  
[FIXED 150](#)  
[FLOOR 148](#)  
[FORECAST 151](#)  
[FV 146](#)  
[GEOMEAN 151](#)  
[GROWTH 151](#)  
[HARMEAN 151](#)  
[HLOOKUP 147](#)  
[HOUR 145](#)  
[HYPERLINK 147](#)  
[IF 147](#)  
[INDEX 147](#)  
[INT 148](#)  
[INTERCEPT 151](#)  
[IPMT 146](#)  
[IRR 146](#)  
[ISERR 146](#)  
[ISERROR 146](#)  
[ISNA 146](#)  
[ISPMT 146](#)  
[KURT 151](#)  
[LARGE 151](#)  
[LEFT 150](#)  
[LEN 150](#)  
[LINEST 151](#)  
[LN 148](#)  
[LOG 148](#)  
[LOG10 149](#)  
[LOGEST 151](#)  
[LOOKUP 147](#)  
[LOWER 150](#)  
[MATCH 147, 152](#)  
[MAX 152](#)  
[MAXA 152](#)  
[MEDIAN 152](#)  
[MID 150](#)  
[MIN 152](#)  
[MINA 152](#)  
[MINUTE 145](#)  
[MIRR 146](#)  
[MOD 149](#)  
[MODE 152](#)  
[MONTH 145](#)  
[MORMINV 152](#)  
[NA 146](#)  
[NEGBINOMDIST 152](#)  
[NORMDIST 152](#)  
[NORMSDIST 152](#)  
[NORMSINV 152](#)  
[NOT 147](#)  
[NOW 145](#)  
[NPER 146](#)  
[NPV 146](#)  
[ODD 149](#)  
[OFFSET 147](#)  
[OR 147](#)  
[PEARSON 152](#)

supported Excel functions (*continued*)

[PERMUT 152](#)  
[PI 149](#)  
[PMT 146](#)  
[POWER 149](#)  
[PPMT 146](#)  
[PRODUCT 149](#)  
[PROPER 150](#)  
[PV 146](#)  
[RADIANT 149](#)  
[RAND 149](#)  
[RATE 146](#)  
[REPLACE 150](#)  
[REPT 150](#)  
[RIGHT 150](#)  
[ROMAN 149](#)  
[ROUND 149](#)  
[ROUNDDOWN 149](#)  
[ROUNDUP 149](#)  
[ROW 147](#)  
[ROWS 148](#)  
[RSQ 152](#)  
[SEARCH 150](#)  
[SECOND 145](#)  
[SIGN 149](#)  
[SIN 149](#)  
[SINH 149](#)  
[SKEW 152](#)  
[SLN 146](#)  
[SLOPE 152](#)  
[SMALL 152](#)  
[SQRT 149](#)  
[STANDARDIZE 152](#)  
[STDEV 152](#)  
[STDEVA 152](#)  
[STDEVP 152](#)  
[STDEVPA 152](#)  
[STEYX 153](#)  
[SUBSTITUTE 150](#)  
[SUM 149](#)  
[SUMIF 149](#)  
[SYD 146](#)  
[T 150](#)  
[TAN 149](#)  
[TANH 149](#)  
[TEXT 150](#)  
[TIME 145](#)  
[TIMEVALUE 145](#)  
[TODAY 145](#)  
[TREND 153](#)  
[TRIM 150](#)  
[TRUE 147](#)  
[UPPER 150](#)  
[VALUE 150](#)  
[VAR 153](#)  
[VARA 153](#)  
[VARP 153](#)  
[VARPA 153](#)  
[VLOOKUP 148](#)  
[WEEKDAY 145](#)  
[WEIBULL 153](#)  
[YEAR 145](#)

## T

### TM1 Web

- administering [69](#)
- administrator tasks [6](#)
- browsing and analyzing data [6](#)
- configuration parameters [71](#)
- homepage [78](#)
- logging [87](#), [88](#)
- logging in [5](#)
- overview [5](#)
- scorecarding [61](#)
- starting [5](#)
- startup parameters [82](#)
- tm1web.log file [87](#)
- using [6](#)

### TM1 Web API [91](#)

TM1 Web JavaScript library, *See* JavaScript library

TM1 Web URL API, *See* URL API

### tm1web\_config.xml

- defined [70](#)
- editing [77](#)
- startup parameters [82](#)

### toolbars

- Cube Viewer [26](#)
- paging [27](#)
- Subset Editor [42](#)
- websheet [12](#)

Translating objects in websheets [12](#)

## U

### unsupported Excel functions

- ACCRINT [156](#)
- ACCRINTM [156](#)
- AMORDEGRC [156](#)
- AMORLINC [156](#)
- AREAS [158](#)
- ASC [161](#)
- BAHTTEXT [161](#)
- BETADIST [160](#)
- BETAINV [160](#)
- CHIDIST [160](#)
- CHIINV [160](#)
- CHITEST [160](#)
- COUNTBLANK [160](#)
- COUNTIFS [160](#)
- COUPDAYBS [156](#)
- COUPDAYS [156](#)
- COUPDAYSNC [156](#)
- COUPNCD [156](#)
- COUPNUM [156](#)
- COUPPCD [156](#)
- CRITBINOM [160](#)
- CUMIPMT [156](#)
- CUMPRINC [156](#)
- DAVERAGE [155](#)
- DCOUNT [155](#)
- DCOUNTA [155](#)
- DGET [155](#)
- DISC [156](#)
- DMAX [155](#)
- DMIN [155](#)
- DOLLARDE [157](#)

### unsupported Excel functions (*continued*)

- DOLLARFR [157](#)
- DPRODUCT [155](#)
- DSTDEV [155](#)
- DSTDEVP [155](#)
- DSUM [155](#)
- DURATION [157](#)
- DVAR [155](#)
- DVARP [155](#)
- EDATE [155](#)
- EFFECT [157](#)
- EOMONTH [155](#)
- ERROR.TYPE [158](#)
- FACTDOUBLE [159](#)
- FDIST [160](#)
- FINV [160](#)
- FREQUENCY [160](#)
- FTEST [160](#)
- FVSCHEDULE [157](#)
- GAMMADIST [160](#)
- GAMMAINV [160](#)
- GAMMALN [160](#)
- GCD [159](#)
- HYPGEOMDIST [160](#)
- IFERROR [158](#)
- INDIRECT [158](#)
- INFO [158](#)
- INTRATE [157](#)
- ISBLANK [158](#)
- ISEVEN [158](#)
- ISLOGICAL [158](#)
- ISNONTEXT [158](#)
- ISNUMBER [158](#)
- ISODD [158](#)
- ISREF [158](#)
- ISTEXT [158](#)
- JIS [161](#)
- LCM [159](#)
- LOGINV [160](#)
- LOGNORMDIST [160](#)
- MDETERM [159](#)
- MDURATION [157](#)
- MINVERSE [159](#)
- MMULT [159](#)
- MROUND [159](#)
- MULTINOMIAL [159](#)
- N [158](#)
- NEGBINOMDIST [160](#)
- NETWORKDAYS [155](#)
- NOMINAL [157](#)
- ODDFPRICE [157](#)
- ODDFYIELD [157](#)
- ODDLPRICE [157](#)
- ODDLYIELD [157](#)
- PERCENTILE [160](#)
- PERCENTRANK [160](#)
- PHONETIC [161](#)
- POISSON [160](#)
- PRICE [157](#)
- PRICEDISC [157](#)
- PRICEMAT [157](#)
- PROB [161](#)
- QUARTILE [161](#)
- QUOTIENT [159](#)

unsupported Excel functions (*continued*)

- [RANDBETWEEN 159](#)
- [RANK 161](#)
- [RECEIVED 157](#)
- [RTD 158](#)
- [SERIESSUM 159](#)
- [SQRTPI 159](#)
- [SUBTOTAL 159](#)
- [SUMISF 159](#)
- [SUMPRODUCT 159](#)
- [SUMSQ 159](#)
- [SUMX2MY2 159](#)
- [SUMX2PY2 159](#)
- [SUMXMY2 159](#)
- [TBILLEQ 157](#)
- [TBILLPRICE 157](#)
- [TBILLYIELD 157](#)
- [TDIST 161](#)
- [TINV 161](#)
- [TRANSPOSE 159](#)
- [TRIMMEAN 161](#)
- [TRUNC 159](#)
- [TTEST 161](#)
- [TYPE 158](#)
- [VDB 157](#)
- [WEEKNUM 155](#)
- [WORKDAY 155](#)
- [XIRR 157](#)
- [XNPV 157](#)
- [YEARFRAC 155](#)
- [YIELD 158](#)
- [YIELDDISC 158](#)
- [YIELDMAT 158](#)
- [ZTEST 161](#)

URL API

- [Action parameter 103](#)
- [AdminHost parameter 101](#)
- [applying actions to objects 104](#)
- [base URL 99](#)
- [basic concepts 100](#)
- [CubeViewer chart type 109](#)
- [CubeViewer charts 109](#)
- [CubeViewer display modes 109](#)
- [CubeViewer display properties 107](#)
- [Cubeviewer title elements 108](#)
- [displaying CubeViewer objects 107](#)
- [displaying websheet objects 105](#)
- [form-based login 102](#)
- [getting started 98](#)
- [HTML iframe 100](#)
- [LegacyUrlApiSessionDiscoveryEnabled parameter 93](#)
- [logging out 103](#)
- [Open parameter 103](#)
- [opening CubeViewer objects 107](#)
- [opening websheet objects 105](#)
- [overview 97](#)
- [parameter reference 111](#)
- [parameters 99](#)
- [session token login 91](#)
- [syntax 98](#)
- [TM1Server parameter 101](#)
- [upgrading older URL API projects 110](#)
- [URL escape characters 100](#)
- [user login and logout 101](#)

URL API (*continued*)

- [websheet display properties 105](#)
- [websheet title elements 106](#)

URL API parameters

- [AccessType 111, 115](#)
- [Action 112](#)
- [AdminHost 113](#)
- [AutoRecalc 114](#)
- [ChartType 114](#)
- [Cube 115](#)
- [HideDimensionBar 116](#)
- [HideToolbar 116](#)
- [TM1Server 116](#)
- [TM1SessionId 117](#)
- [Type 117](#)
- [View 117](#)
- [Workbook 118](#)

user-defined consolidations

- [See custom consolidations 49](#)

**V**

views

- [creating 36](#)

**W**

Web charts

- [chart type 39](#)
- [drill through 39](#)

[web.config 77](#)

websheet objects

- [with JavaScript library 122](#)
- [with URL API 105](#)

websheet properties

- [admin host 20](#)
- [changing 20](#)
- [dimension list 20](#)
- [general 20](#)
- [print 20](#)
- [write back 20](#)

worksheets

- [cell protection 11](#)
- [conditional formatting 10](#)
- [defined 9](#)
- [diagonal borders 9](#)
- [display of gridlines 9](#)
- [editing data in cells 13](#)
- [editing data overview 13](#)
- [freeze panes 11](#)
- [hide columns 10](#)
- [hyperlinks 10](#)
- [making read-only 20](#)
- [relational drill 9](#)
- [spreading data 14](#)
- [toolbar 12](#)
- [visual differences with Excel worksheets 9](#)

Workbook class

- [methods 134](#)
- [properties 129](#)
- [wrapping in cells 85](#)
- [write back 20, 31](#)



