# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal Information</td>
<td>5</td>
</tr>
<tr>
<td><strong>1. Introduction</strong></td>
<td>7</td>
</tr>
<tr>
<td>1.1. Introducing Intel Trace Analyzer</td>
<td>7</td>
</tr>
<tr>
<td>1.2. What's New</td>
<td>7</td>
</tr>
<tr>
<td>1.3. Notational Conventions</td>
<td>8</td>
</tr>
<tr>
<td>1.4. Related Information</td>
<td>9</td>
</tr>
<tr>
<td>1.5. Starting Intel(R) Trace Analyzer</td>
<td>9</td>
</tr>
<tr>
<td>1.5.1. Starting in UNIX* Environment</td>
<td>9</td>
</tr>
<tr>
<td>1.5.2. Starting in Windows* Environment</td>
<td>10</td>
</tr>
<tr>
<td>1.5.3. Trace Cache</td>
<td>10</td>
</tr>
<tr>
<td>1.5.4. Command Line Interface (CLI)</td>
<td>10</td>
</tr>
<tr>
<td>1.6. Internationalization</td>
<td>10</td>
</tr>
<tr>
<td><strong>2. Welcome Page</strong></td>
<td>13</td>
</tr>
<tr>
<td><strong>3. Main Menu</strong></td>
<td>14</td>
</tr>
<tr>
<td>3.1. File Menu</td>
<td>14</td>
</tr>
<tr>
<td>3.2. Options Menu</td>
<td>15</td>
</tr>
<tr>
<td>3.2.1. Preferences</td>
<td>15</td>
</tr>
<tr>
<td>3.2.2. General Preferences</td>
<td>15</td>
</tr>
<tr>
<td>3.2.3. Tracefile Preferences</td>
<td>17</td>
</tr>
<tr>
<td>3.2.4. Event Timeline Settings</td>
<td>17</td>
</tr>
<tr>
<td>3.2.5. Qualitative Timeline Settings</td>
<td>19</td>
</tr>
<tr>
<td>3.2.6. Quantitative Timeline Settings</td>
<td>20</td>
</tr>
<tr>
<td>3.2.7. Counter Timeline Settings</td>
<td>22</td>
</tr>
<tr>
<td>3.2.8. Function Profile Settings</td>
<td>25</td>
</tr>
<tr>
<td>3.2.9. Message Profile Settings</td>
<td>27</td>
</tr>
<tr>
<td>3.2.10. Collective Operations Profile Settings</td>
<td>34</td>
</tr>
<tr>
<td>3.3. Font Settings</td>
<td>37</td>
</tr>
<tr>
<td>3.4. Number Formatting Settings</td>
<td>38</td>
</tr>
<tr>
<td>3.5. Project Menu</td>
<td>39</td>
</tr>
<tr>
<td>3.6. Windows Menu</td>
<td>40</td>
</tr>
<tr>
<td>3.7. Help Menu</td>
<td>41</td>
</tr>
<tr>
<td><strong>4. Views</strong></td>
<td>42</td>
</tr>
<tr>
<td>4.1. View Main Menu</td>
<td>42</td>
</tr>
<tr>
<td>4.1.1. View Menu</td>
<td>42</td>
</tr>
<tr>
<td>4.1.2. Charts Menu</td>
<td>44</td>
</tr>
<tr>
<td>4.1.3. Navigate Menu</td>
<td>45</td>
</tr>
<tr>
<td>4.1.4. Advanced Menu</td>
<td>46</td>
</tr>
<tr>
<td>4.1.5. Layout Menu</td>
<td>49</td>
</tr>
<tr>
<td>4.1.6. Comparison Menu</td>
<td>51</td>
</tr>
<tr>
<td>4.2. Toolbar</td>
<td>51</td>
</tr>
<tr>
<td>4.3. Trace Map</td>
<td>52</td>
</tr>
<tr>
<td>4.4. Status Bar</td>
<td>54</td>
</tr>
<tr>
<td>4.5. Navigation in Time</td>
<td>54</td>
</tr>
<tr>
<td>4.5.1. Zoom Stack</td>
<td>55</td>
</tr>
<tr>
<td><strong>5. Charts</strong></td>
<td>58</td>
</tr>
<tr>
<td>5.1. Event Timeline</td>
<td>58</td>
</tr>
<tr>
<td>5.1.2. Context Menu</td>
<td>60</td>
</tr>
</tbody>
</table>
6.12. Time Interval Selection ................................................................. 136
6.13. Configuration Dialog Boxes ...................................................... 137
   6.13.1. Load Configuration Dialog .............................................. 137
   6.13.2. Edit Configuration Dialog .............................................. 137
6.14. Find Dialog Box ........................................................................ 138

7. Correctness Checking Reports ...................................................... 141
   7.1. Appearance .............................................................................. 141
       7.1.1. Event Timeline Correctness Checking Reports .............. 141
       7.1.2. Qualitative Timeline Correctness Checking Reports ...... 142
       7.1.3. Detailed Dialog .............................................................. 143
   7.2. Caching .................................................................................. 144

8. Comparison of two Trace Files .................................................... 145
   8.2. Mappings in Comparison Views .............................................. 146
       8.2.2. Mapping of Processes ..................................................... 147
       8.2.3. Mapping of Functions ...................................................... 148
   8.3. Comparison Charts ................................................................. 149
       8.3.1. Comparison Function Profile ....................................... 149
       8.3.2. Comparison Message Profile ....................................... 151
       8.3.3. Comparison Collective Operations Profile ..................... 153

9. Command Line Interface (CLI) ...................................................... 155

10. Custom Plug-in Framework ......................................................... 161
    10.1. Usage Instructions ............................................................... 161
    10.2. Developing Simulators ........................................................ 161
        10.2.1. Building a Simulator Library .................................... 162
        10.2.2. Writing your Simulator .............................................. 162
        10.2.3. CPF Simulator API ...................................................... 163

11. Intel® Trace Collector Configuration Assistant .............................. 171
    11.1. General Description ............................................................ 171
    11.2. Detailed Description ............................................................ 171

12. Concepts ..................................................................................... 174
    12.1. Level of Detail ................................................................. 174
    12.2. Aggregation ........................................................................ 175
        12.2.1. Function Aggregation ............................................... 175
        12.2.2. Thread Aggregation .................................................... 176
    12.3. Advanced Aggregation ...................................................... 176
    12.4. Tagging and Filtering ........................................................ 179
        12.4.1. Tagging ................................................................. 179
        12.4.2. Filtering ............................................................... 179
LEGAL INFORMATION

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL’S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS’ FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to:
http://www.intel.com/design/literature.htm


1. Introduction

This document describes the feature set of the Intel® Trace Analyzer. See the following topics:

- Introducing Intel® Trace Analyzer
- What's New
- Notational Conventions
- Related Information

1.1. Introducing Intel Trace Analyzer

Intel® Trace Analyzer is a graphical tool that displays and analyzes event trace data generated by the Intel® Trace Collector.

Intel® Trace Analyzer helps to detect performance problems, programming errors and understand the behavior of the application.

Intel® Trace Analyzer

1.2. What’s New

This reference guide documents the Intel® Trace Analyzer 8.1 Update 4.
Intel® Trace Analyzer 8.1 Update 4:
- This release contains minor improvements and bug-fixes.

Intel® Trace Analyzer 8.1 Update 3:
- New Trace Map was added. For details, see Trace Map.
- The Preferences dialog box now contains new tabs for timeline settings. For more information, see Preferences.

Intel® Trace Analyzer 8.1 Update 2:
- New Toolbar was added. It provides quick access to the Intel Trace Analyzer functionality. For details, see Toolbar.
- New Event Timeline preferences tab was added to the Preferences dialog box. For more information, see Preferences.

Intel® Trace Analyzer 8.1 Update 1:
- New Welcome Page was added. It provides a quick start to operating Intel Trace Analyzer. For details, see Welcome Page.
- New Preferences dialog was added. It enables you to set up your working environment. See, Preferences.

1.3. Notational Conventions

The following conventions are used in this document.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>This type style</td>
<td>Introduces new terms, denotation of terms, placeholders, or titles of manuals.</td>
<td>The term process in this documentation implicitly includes thread.</td>
</tr>
<tr>
<td>This type style</td>
<td>Denotes GUI elements</td>
<td>Click OK</td>
</tr>
<tr>
<td>This type style</td>
<td>Hyperlinks</td>
<td><a href="https://premier.intel.com/">https://premier.intel.com/</a></td>
</tr>
<tr>
<td>This type style</td>
<td>Commands, arguments, options</td>
<td>traceanalyzer --cli trace.stf -c0 -w</td>
</tr>
<tr>
<td>&lt;this type style&gt;</td>
<td>Placeholders for actual values</td>
<td>&lt;new_name&gt;</td>
</tr>
<tr>
<td>$</td>
<td>Introduces UNIX* commands</td>
<td>$ ls</td>
</tr>
<tr>
<td>&gt;</td>
<td>Indicates a menu item inside a menu</td>
<td>Main Menu &gt; File &gt; Quit</td>
</tr>
</tbody>
</table>
NOTE

The term *process* in this documentation implicitly includes thread. As soon as the Intel® Trace Analyzer loads a trace file that was generated running a multithreaded application, the GUI uses the term *thread* only if it is applicable. This is done to avoid confusing MPI application programmers who normally use the term *process* instead of thread.

1.4. Related Information


Intel® Premier Customer Support is available at: [https://premier.intel.com/](https://premier.intel.com/)


1.5. Starting Intel(R) Trace Analyzer

This topic describes the following scenarios to start the Intel® Trace Analyzer:

- **Starting in UNIX* environment**
- **Starting in Windows* environment**
- **Trace cache**
- **Command line interface (CLI)**

**NOTE**

For a proper functioning of the Intel Trace Analyzer, ensure that files do not get modified while they are opened in the Intel Trace Analyzer.

1.5.1. Starting in UNIX* Environment

In UNIX* environment, invoke the Intel® Trace Analyzer through the command line:

```
$ traceanalyzer
```

Optionally, specify one or more trace files as arguments. For example:

```
$ traceanalyzer poisson_icomm.single.stf &
```

The above example opens the trace file `poisson_icomm.single.stf` in the Intel Trace Analyzer.

To open trace files in the Intel Trace Analyzer without restarting it, go to **File > Open.**
1.5.2. Starting in Windows* Environment

In Windows* environment, double-click on a trace file to invoke the Intel® Trace Analyzer or use the menu item Start > All Programs > Intel® Software Development Products > Intel® Trace Analyzer and Collector 8.1 Update 4 > Intel® Trace Analyzer.

1.5.3. Trace Cache

For every opened trace file, Intel® Trace Analyzer creates a trace cache. It is stored in the same directory as the trace file and has the same file name but with suffix .cache added.

1.5.4. Command Line Interface (CLI)

Intel® Trace Analyzer provides a command line interface (CLI) in UNIX* and Windows* environments.

Use the CLI to:

- Pre-calculate trace caches for new trace files from batch files without invoking the graphical user interface. Use it for very big trace files.
- Automate profiling data producing for several trace files without further interaction.

See Also
Command Line Interface (CLI)

1.6. Internationalization

When it comes to internationalization, the GUI is as agnostic as possible.

There is only English version of the software and the documentation. The number formats in the GUI and in the exported text files always use a decimal point to separate three digit groups, and a space character is used instead of a comma or point.

When moving trace files between different systems and sharing them among people, use pure ASCII characters in path and file names as much as possible.

On Windows* systems:

Since Windows* OS uses the same encoding (UTF-16) in file and path names regardless of your language and region settings, you do not need to configure anything to make Intel® Trace Analyzer work in your environment. As soon as the file names appear correctly in the folder listings, the Intel Trace Analyzer can use them.

On Linux* systems:

To be able to open your trace file, set the encoding part in your locale settings to the UTF-8 encoding, because it is space efficient, can express nearly all languages, is supported on most systems, and is identical to 7bit ASCII encoding for codes 0 to 127. If you use plain ASCII in your path names, you do not need to configure anything.

Valid locale settings look like en_US.latin-1 for U.S., de_DE.UTF-8 for Germany, fr_CA.iso88591 for French-speaking people in Canada and ja JP.ujis for Japan. The command locale helps to list all available locales and encodings on your system. For Intel Trace
Analyzer, only the encoding part of the locale setting is relevant. All file names and path names given on the command line and read from the file system are expected to be encoded in the encoding given by the locale settings.

**NOTE**

Ensure that you use the same encoding for the names of folders and files stored in these folders. Different encodings of folder and file names can lead to inaccessible files.

Intel Trace Analyzer supports the following encodings (but use the encoding names as given by `locale -m`):

1. Latin1
2. Big5 - Chinese
3. Big5-HKSCS - Chinese
4. eucJP - Japanese
5. eucKR - Korean
6. GB2312 - Chinese
7. GBK - Chinese
8. GB18030 - Chinese
9. JIS7 - Japanese
10. Shift-JIS - Japanese
11. TSCII - Tamil
12. utf8 - Unicode, 8-bit
13. utf16 - Unicode
14. KOI8-R - Russian
15. KOI8-U - Ukrainian
16. ISO8859-1 - Western
17. ISO8859-2 - Central European
18. ISO8859-3 - Central European
19. ISO8859-4 - Baltic
20. ISO8859-5 - Cyrillic
21. ISO8859-6 - Arabic
22. ISO8859-7 - Greek
23. ISO8859-8 - Hebrew, visually ordered
24. ISO8859-8-i - Hebrew, logically ordered
25. ISO8859-9 - Turkish
26. ISO8859-10
27. ISO8859-13
28. ISO8859-14
29. ISO8859-15 - Western
30. IBM 850
31. IBM 866
32. CP874
33. CP1250 - Central European
34. CP1251 - Cyrillic
35. CP1252 - Western
36. CP1253 - Greek
37. CP1254 - Turkish
38. CP1255 - Hebrew
39. CP1256 - Arabic
40. CP1257 - Baltic
41. CP1258
42. TIS-620 - Thai

See Also
General Preferences
Number Formatting Settings
2. Welcome Page

Welcome Page provides a quick start to operating the Intel® Trace Analyzer.

From the Welcome Page you can open:

<table>
<thead>
<tr>
<th>Press</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open...</td>
<td>Tracefile/Project dialog box</td>
</tr>
<tr>
<td>Open Samples</td>
<td>Intel Trace Analyzer sample tracefiles</td>
</tr>
<tr>
<td>Intel® Trace Analyzer Help</td>
<td>Intel® Trace Analyzer Reference Guide</td>
</tr>
<tr>
<td>Getting Started Guide</td>
<td>Getting Started Guide that describes the essential steps to start working with the Intel® Trace Analyzer</td>
</tr>
<tr>
<td>Tutorial</td>
<td>Tutorial on detecting and removing unnecessary serialization in your application</td>
</tr>
<tr>
<td>Recent Traces</td>
<td>Recently opened tracefiles</td>
</tr>
</tbody>
</table>

You can also find Open Trace menu and recently opened tracefiles in the File Menu and Intel® Trace Analyzer Help in the Help Menu.

See Also
File Menu
Help Menu
3. Main Menu

Use the Intel® Trace Analyzer main menu options to change the configuration settings and the display style of the Intel Trace Analyzer.

There are five sub-menus in the main menu:

- **File Menu**
- **Options Menu**
- **Project Menu**
- **Windows Menu**
- **Help Menu**

3.1. File Menu

The File menu has the following entries: **Open**, **Open Samples**, **Load Configuration**, **Edit Configuration** and **Quit**.

<table>
<thead>
<tr>
<th>Select This:</th>
<th>Shortcut</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open...</td>
<td>Ctrl+O</td>
<td>Open one or more trace files</td>
</tr>
<tr>
<td>Open Samples...</td>
<td>Ctrl+S</td>
<td>Open sample trace files</td>
</tr>
<tr>
<td>Load Configuration</td>
<td>none</td>
<td>Select a new configuration</td>
</tr>
<tr>
<td>Edit Configuration</td>
<td>none</td>
<td>Edit configuration, for example, by changing values or removing entries from the configuration.</td>
</tr>
<tr>
<td>Quit</td>
<td>Ctrl+Q</td>
<td>Exit the Intel® Trace Analyzer</td>
</tr>
</tbody>
</table>

The list above the Quit entry displays up to ten recently opened trace files.
3.2. Options Menu

The Options Menu has the following entries: Preferences, Set Fonts and Number Formatting.

Use the Preferences dialog box to set up your working environment.

Use the Set Fonts option to change the fonts for the individual Charts.

In the Number Formatting dialog box, specify the format of the various numerical values displayed in the Intel Trace Analyzer.

3.2.1. Preferences

Use the Preferences option to set up your working environment.

This section describes the following preference dialog boxes:

- General Preferences
- Tracefile Preferences
- Event Timeline Settings
- Qualitative Timeline Settings
- Quantitative Timeline Settings
- Counter Timeline Settings

3.2.2. General Preferences

The General Preferences tab enables you to set up the basic Intel® Trace Analyzer behavior, such as the startup options, view style, confirmation before quitting the Intel Trace Analyzer and the number of the recently opened tracefiles.
Configure the Intel Trace Analyzer behavior in the following sections of the General Preferences dialog box:

<table>
<thead>
<tr>
<th>Section</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>On Startup</strong></td>
<td>Choose the file, which opens at the Intel Trace Analyzer startup. It can be the <strong>Welcome Page</strong>, the last tracefile you worked with or any other tracefile that you can specify. In this section you can also enable/disable the <strong>Splash Screen</strong> appearance and the main window maximization.</td>
</tr>
<tr>
<td><strong>Appearance</strong></td>
<td>Choose the design in which the view is displayed. The options in the <strong>Style</strong> section depend on the environment. If the default style does not appeal to you, choose another view design from the drop-down list.</td>
</tr>
<tr>
<td><strong>Other Options</strong></td>
<td>Customize the number of the recent tracefiles to appear in the <strong>File Menu</strong>. You can also set the application to ask confirmation before you quit the Intel Trace Analyzer.</td>
</tr>
</tbody>
</table>
3.2.3. Tracefile Preferences

The **Tracefile Preferences** tab enables you to set up tracefile options. Use the tab to specify the chart with which a new tracefile or a new view of the tracefile should be opened. By default, it is the Function Profile chart.

![Preferences dialog box](image)

3.2.4. Event Timeline Settings

Use the **Event Timeline Settings** in the Preferences dialog box to change the display settings, the layout settings and the colors of the Event Timeline Chart.
### Option | Result
---|---
**Minimal Spacing Between Bars** | Adjust the space between the function bars. The unit is pixel.
**Minimal Bar Height slider** | Adjust the height of the function bar itself. The unit is pixel.
**Adjust Minimal Bar Height to Labels** | Check this box to make the bars tall enough to display a function label. This option is checked by default.
**Use Available Vertical Space** | Check this box to influence the overall vertical layout of the bars. This option is checked by default.
**Function Colors** | Change the color in which functions are represented. When you press this button, the Function Group Color Editor is called up.
**Message Color** | Change the color of messages
**Collective Operation Color** | Change the color of Collective Operations
The colors that you choose for messages and collective operations are local to the Event Timeline Chart. But the colors you choose for functions or function groups are shared by all Charts and Views in the same tracefile.

See Also
Function Group Color Editor

3.2.5. Qualitative Timeline Settings

The Qualitative Timeline Settings dialog box consists of a Display group with check boxes and a Vertical Scaling combo-box. Use the combo-box to adjust the vertical scaling of the timeline.

In the Display group, specify which scales to show and whether to display a timeline legend. By default, the vertical scale and the legend are shown.

Use the Vertical Scaling group to switch between the default Automatic Scaling and the Manual Scaling of the y-axis. To specify explicitly the maximum scale value, use Manual Scaling. To compare visually two or more Charts in the same or distinct Views, specify the same maximum value for the charts.
3.2.6. Quantitative Timeline Settings

Use the **Quantitative Timeline Settings** dialog box to adjust the display options and the scaling of the Qualitative Timeline Chart.

Qualitative Timeline Settings dialog box has a **Display** group and a **Vertical Scaling** group. In the **Display** group, there are the following controls:

<table>
<thead>
<tr>
<th>Option</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Scale</td>
<td>Display a time scale along the x-axis</td>
</tr>
<tr>
<td>Vertical Scale</td>
<td>Show the scale on the Y-axis</td>
</tr>
<tr>
<td>Legend</td>
<td>Show the legend in the right margin of the timeline.</td>
</tr>
<tr>
<td>Adjust graphics to legend height</td>
<td>Make the size of the diagram large enough to show all legend items</td>
</tr>
<tr>
<td>Frames</td>
<td>Frames give an outline to the bars. The frames become visible only when you zoom in very closely so that each bar is separated from the other.</td>
</tr>
</tbody>
</table>
Grid

Draw the grid on top of the data and align it with the ticks on the scales

Vertical Scaling

This option functions the same way as in the Qualitative Timeline

Granularity

Define the level of detail you want the Quantitative Timeline to display. It may be either Individual processes/Threads or just top level entries of Group_All_Processes.

Check the **Frames** box to outline the bars in the Quantitative Timeline:

Uncheck the **Grid** box:
3.2.7. Counter Timeline Settings

The Counter Timeline Settings dialog box has two tabs: the Preferences tab and the Counters tab.

3.2.7.1. Preferences Tab

Use the controls in this tab to adjust the display settings and the Counter Timeline scaling. The tab consists of the groups Display and Vertical Scaling.
In the **Display** group, configure the Counter Timeline with the following controls:

<table>
<thead>
<tr>
<th>Check This Box:</th>
<th>If You Want the Counter Timeline to:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time Scale</strong></td>
<td>Display the time scale along the x-axis.</td>
</tr>
<tr>
<td><strong>Vertical Scale</strong></td>
<td>Display the time scale along the y-axis. By default, this option is enabled so that you can see the scale.</td>
</tr>
<tr>
<td><strong>Legend</strong></td>
<td>Show the legend in the right margin of the timeline. It is also enabled by default.</td>
</tr>
<tr>
<td><strong>Adjust graphics to legend height</strong></td>
<td>Make the size of the diagram large enough to show all legend items.</td>
</tr>
<tr>
<td><strong>Single Diagram/Many Diagrams combo-box</strong></td>
<td>Select either a single diagram for all selected counters and all selected target group children or a separate diagram for each target group child.</td>
</tr>
</tbody>
</table>

Use the **Vertical Scaling** group to switch between the **Automatic Scaling (per Diagram)**, **Automatic Scaling (Global)** and the **Manual Scaling** of the y-axis. To specify explicitly the
minimum and maximum scale values, choose **Manual Scaling**. To compare visually two or more Charts in the same or distinct Views, specify the same bounds for both charts.

### 3.2.7.2. Counters Tab

This tab contains seven entries for each trace file counter. The top level entry for each counter shows the counter Name, its Unit, its target Process Group and its Attributes. The name and the unit are arbitrary, free-format strings defined in the trace file.

![Counters Tab Image]

The column **Process Group** contains the counter target group. A counter that has different values for each process has the target group `All_Processes`, while a counter that has a distinct value for each SMP node belongs to the target group `All_Nodes`.

The column **Attributes** can contain either the Integer or Double attribute, indicating the counter type.

One of **Valid BEFORE Point**, **Valid AFTER Point**, **Valid AT Point** or **Curve** indicates the counters scope. Curve indicates that it is meaningful to interpolate values between two given counter values.

The attribute **Show Rate** indicates that it is preferable to display the derivation to time instead of the plain counter values.

Under the top-level counter entry, there are six entries for the minimum, average and maximum values and for the minimum, average and maximum rates, which you can switch on and off.
independently. If you just use the top-level entries check box, then either the average value or average rate is chosen depending on the counter attributes.

### 3.2.8. Function Profile Settings

The Function Profile Settings dialog box enables you to customize displayed options for all the four tabs of the Function Profile Chart. To access the Preferences dialog box, right click on the chart and select Function Profile Settings from the context menu.

#### 3.2.8.1. Preferences Tab

There are four groups of options in the Preferences tab. Use this tab to set up the display, time, scale bars and colors.

**Preferences Tab of the Function Profile Settings**

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>View: C:/Working/ITA/nv_itac_20130228_ia12/examples/traces/boisson_jcomm.single.stf</td>
<td></td>
</tr>
<tr>
<td>Chart: # Function Profile</td>
<td></td>
</tr>
<tr>
<td><strong>Preferences</strong> tab:</td>
<td></td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td></td>
</tr>
<tr>
<td>Time Self</td>
<td>✓</td>
</tr>
<tr>
<td>Time Total</td>
<td>✓</td>
</tr>
<tr>
<td>#Calls</td>
<td>✓</td>
</tr>
<tr>
<td>Time Self per Call</td>
<td>✓</td>
</tr>
<tr>
<td>Time Total per Call</td>
<td>✓</td>
</tr>
<tr>
<td>#Processes</td>
<td></td>
</tr>
<tr>
<td>Time Self per Process</td>
<td></td>
</tr>
<tr>
<td>Time Total per Process</td>
<td></td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td></td>
</tr>
<tr>
<td>100% == Time Interval</td>
<td></td>
</tr>
<tr>
<td>0% == Time Interval * #Processes</td>
<td></td>
</tr>
<tr>
<td><strong>Scale Bars to Maximum of</strong></td>
<td></td>
</tr>
<tr>
<td>All Items</td>
<td></td>
</tr>
<tr>
<td>Visible Items</td>
<td></td>
</tr>
<tr>
<td>Siblings</td>
<td></td>
</tr>
<tr>
<td><strong>Colors</strong></td>
<td></td>
</tr>
<tr>
<td>Function Colors</td>
<td></td>
</tr>
</tbody>
</table>
### Time Self
Display time spent in the given function, excluding time spent in functions called from it.

### Time Total
Display time spent in the given function, including time spent in functions called from it.

### #Calls
Display the number of calls to this function.

If other attributes are non-zero, this attribute can be zero, because the actual calls to the respective function can occur outside the current time interval.

### Time Self per Call
Display \text{Time Self} averaged over \#Calls.

### Time Total per Call
Display \text{Time Total} averaged over \#Calls.

### #Processes
Display the number of processes in this function.

### Time Self per Process
Display \text{Time Self} averaged over \#Processes.

### Time Total per Process
Display \text{Time Total} averaged over \#Processes.

### Function Colors
Open the \text{Function Group Color Editor}.

Use the given check boxes to switch on and off the displaying of the above attributes either as text or as a bar graph. By default, only four boxes are checked - \text{Time Self}, \text{Time Total}, \#Calls, and \text{Time Self per Call}.

Use the radio buttons to specify the time format (seconds or ticks) or to specify time as the percentage of the time interval.

The scaling modes are given as radio buttons. They are:

- The default \text{Visible Items} scales the bars to the respective maximum of all expanded items.
- \text{All Items} uses the global maximum of all values, regardless of whether they are expanded or not.
- \text{Siblings} uses only the maximum of the direct siblings.

All the three scaling modes use values only from the same column.

### 3.2.8.2. Processes Tab

If you want particular process to be displayed in the Function Profile:

1. Go to the \text{Processes} tab of the Function Profile Settings and choose the necessary processes.

2. Choose the \text{As selected in Settings} option from the combo box.

To select all but one process:

1. Choose the process you do not need.

2. Using the \text{Invert All} option to reverse the selection.
NOTE

It does not influence the current process group of the View, but only focuses the Function Profile on a subset of all processes.

3.2.8.3. Pies Tab

Switch the individual diagram titles and the global legend on and off in the Pies tab.

See Also
Function Group Color Editor

3.2.9. Message Profile Settings

The Settings dialog box contains three tabs: Preferences, Colors and Data.

3.2.9.1. Preferences Tab

Use the Preferences tab to configure Message Profile display and layout.
### Select This: | To Do This:
--- | ---
Row Labels | Switch row headers
Column Labels | Switch column headers
Scale | Switch the colored scale next to the matrix
Grid | Switch the black grid shown between cells on/off
Keep Empty Rows/Columns when using Sender/Receiver Groupings | Switch the special feature on/off. This feature is only relevant for the groupings Sender and Receiver. When you check the box, all processes are shown. For example, even empty rows and columns will be shown. It keeps the form of the matrix constant and makes it easy to look for patterns in the data.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>When you uncheck the box, empty rows and columns are suppressed even in these groupings. All other groupings suppress empty rows and columns to save screen space regardless of the state of this check box.</td>
</tr>
<tr>
<td>Communicator Names</td>
<td>See helpful communicator names (if available in the trace file). It may take a lot of valuable screen space.</td>
</tr>
<tr>
<td>Communicator Ids</td>
<td>Show only concise communicator ids</td>
</tr>
<tr>
<td>Automatic Cell Sizes</td>
<td>Adjust cell sizes to make all text readable</td>
</tr>
<tr>
<td>Equal Cell Sizes</td>
<td>Set equal column widths and enable the check box <strong>Square Cell Sizes</strong></td>
</tr>
<tr>
<td>Square Cell Sizes</td>
<td>Make cells square</td>
</tr>
<tr>
<td>Manual Cell Sizes</td>
<td>Specify the size of the cells in pixels either in the <strong>Cell Size</strong> group at the bottom of the tab or using the slider on top of the matrix. In this mode, the alphanumerical data in the cells is displayed only if it fits or if it is switched off entirely by un-checking the check box <strong>Text in Cells</strong>.</td>
</tr>
</tbody>
</table>

### 3.2.9.2. Colors Tab

In this tab you can change the default colors in which the messages in the Message Profile are shown.
The Message Profile Setting provide the following options:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Color</td>
<td>Choose the colors for the maximum attribute values</td>
</tr>
<tr>
<td>Minimum Color</td>
<td>Choose the colors for the minimum attribute values</td>
</tr>
<tr>
<td>Text input field</td>
<td>Specify the number of color steps (1-255). The chosen colors are considered as points in a color space. The scale colors are interpolated on a line through color space connecting these two points.</td>
</tr>
<tr>
<td>HSV/RGB combo box</td>
<td>Choose between HSV and RGB color space. HSV is more fancy and colorful, but RGB is often more useful and readable.</td>
</tr>
</tbody>
</table>
For monochrome printing, you will get a better result if you choose a very light and a very dark color. For example, white for the minimum and black for the maximum.

**Manual Scaling**

Specify the minimum and maximum values for the color scale in the two text input fields below.

This is very convenient when you compare two Message Profile Charts that may live in different Views.

### 3.2.9.3. Data Tab

The **Grouping** group provides two combo boxes to choose how the data is grouped into categories. The groupings for rows and columns are chosen independently. Note that not all combinations are possible. You cannot set the same grouping for rows and columns and you cannot have Sender/Receiver at one axis and any one of Sender or Receiver on the other axis.

The available groupings are:
## Grouping

<table>
<thead>
<tr>
<th>Grouping</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender</td>
<td>Categorizes the messages by Sender. The exact labels are defined by the current thread group that is given by the View (see the Views section).</td>
</tr>
<tr>
<td>Receiver</td>
<td>Categorizes the messages by Receiver. The exact labels are defined by the current process group that is given by the View.</td>
</tr>
<tr>
<td>Sender/Receiver</td>
<td>Categorizes the messages by Sender/Receiver pairs. The exact labels are defined by the current process group that is given by the View.</td>
</tr>
<tr>
<td>Tag</td>
<td>Categorizes the messages by the MPI tag assigned to the message by the program at the sender side.</td>
</tr>
<tr>
<td>Communicator</td>
<td>Categorizes the messages by the MPI communicator. The labels are either communicator ids or names. Names are displayed if they are available in the trace file and if they are chosen in the Preferences tab of the Message Profile Settings dialog box.</td>
</tr>
<tr>
<td>Volume</td>
<td>Categorizes the messages by their Volume; for example, size in bytes. Grouping Volume by Receiver shows only messages with a volume of 2000 bytes.</td>
</tr>
<tr>
<td>Sending Function</td>
<td>Categorizes the messages by the function that sends them. Labels are names of MPI functions such as MPI_Irsend. This categorization is not influenced by the current Function Aggregation. This information is only available with traces created by the Intel® Trace Collector version 6 and higher.</td>
</tr>
<tr>
<td>Receiving Function</td>
<td>Categorizes the messages by the function that receives them. Labels are names of MPI functions like MPI_Waitany. This is not influenced by the current Function Aggregation. This information is only available with traces created by the Intel® Trace Collector version 6 and higher.</td>
</tr>
<tr>
<td>Datum group</td>
<td>Allows choosing which attribute should be printed or painted in the cells. See the description of the attributes.</td>
</tr>
<tr>
<td>Row Statistics group</td>
<td>Allows switching the individual columns on or off. These</td>
</tr>
</tbody>
</table>
columns hold the statistics for the rows.

**Column Statistics** group

Allows switching the individual rows on or off. These hold the statistics for the columns.

The attributes of the **Datum** group are:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Time</strong></td>
<td>The total travel time of the messages, accumulated over all messages that fall into this cell. The unit is either [s] or [tick] depending on the View setting.</td>
</tr>
<tr>
<td><strong>Minimum Time</strong></td>
<td>The minimum travel time of a message, minimized over all messages that fall into this cell. The unit is either [s] or [tick] depending on the View setting.</td>
</tr>
<tr>
<td><strong>Maximum Time</strong></td>
<td>The maximum travel time of a message, maximized over all messages that fall into this cell. The unit is either [s] or [tick] depending on the View setting.</td>
</tr>
<tr>
<td><strong>Average Transfer Rate, [B/s]</strong></td>
<td>The average transfer rate, averaged over the transfer rates of all messages that fall into this cell. Messages are not weighted; for example, transfer rates of short messages have the same impact as transfer rates of long messages.</td>
</tr>
<tr>
<td><strong>Minimum Transfer Rate, [B/s]</strong></td>
<td>The minimum transfer rate, minimized over all messages that fall into this cell.</td>
</tr>
<tr>
<td><strong>Maximum Transfer Rate, [B/s]</strong></td>
<td>The maximum transfer rate, maximized over all messages that fall into this cell.</td>
</tr>
<tr>
<td><strong>Total Data Volume, [B]</strong></td>
<td>The total data volume, accumulated over all messages that fall into this cell.</td>
</tr>
<tr>
<td><strong>Minimum Data Volume, [B]</strong></td>
<td>The minimum data volume, minimized over all messages that fall into this cell.</td>
</tr>
<tr>
<td><strong>Maximum Data Volume, [B]</strong></td>
<td>The maximum data volume, maximized over all messages that fall into this cell.</td>
</tr>
<tr>
<td><strong>Count, [1]</strong></td>
<td>The number of messages that fall into this cell.</td>
</tr>
</tbody>
</table>
Grouping Volume by Receiver

See Also
Views

3.2.10. Collective Operations Profile Settings

Use the Collective Operations Settings to change the settings of the Collective Operations Profile and the Chart display. You can change the colors, the layout and the statistical attributes.

The Collective Operation Profile Settings dialog box is divided into three tabs namely the Preferences tab, the Colors tab and the Data tab.

3.2.10.1. Preferences Tab

Use the Preferences tab to adjust the Display settings and the Layout settings.

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Labels</td>
<td>Switch the display of the row headers on/off</td>
</tr>
<tr>
<td>Column Labels</td>
<td>Switch the display of the column headers on/off</td>
</tr>
<tr>
<td>Scale</td>
<td>Display the colored scale that is seen on the right-hand side of the matrix</td>
</tr>
<tr>
<td>Grid</td>
<td>Displays/removes the black grid in which the cells are placed</td>
</tr>
<tr>
<td>Keep Empty Rows/Columns when</td>
<td>Switches a special feature that is only relevant for the Groupings Sender and Receiver.</td>
</tr>
</tbody>
</table>
using Sender/Receiver Groupings
For these groupings, check the box to indicate that all processes should always be shown, like for example, showing even empty rows and columns. It keeps the form of the matrix constant and makes it easy to look for patterns in the data.

Uncheck the box to indicate that empty rows and columns even for these groupings are suppressed. All other groupings suppress empty rows and columns to save screen space regardless of the state of this check box.

Communicator Names
See helpful communicator names (if available in the trace file) that may take a lot of valuable screen space

Communicator Ids
Restrict the display to show only concise communicator ids.

Automatic Cell Sizes mode, checking Equal Cell Sizes
Make each column equal in width.
Select it also to enable the check box Square Cell Sizes

Square Cell Sizes
Get square cells.

Manual Cell Sizes
Specify the size of the cells in pixels. This is done either in the Cell Size group at the bottom of the tab or by sizing the cells manually with the slider that is available on top of the matrix as soon as this setting is applied.

In this mode, the alphanumerical data in the cells is displayed only if it fits. Otherwise, it is switched off entirely by unchecking the check box Text in Cells.

3.2.10.2. Colors Tab

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Color</td>
<td>Choose the colors for the maximum attribute values</td>
</tr>
<tr>
<td>Minimum Color</td>
<td>Choose the colors for the minimum attribute values</td>
</tr>
<tr>
<td>Text input field</td>
<td>Specify the number of color steps (1-255)</td>
</tr>
</tbody>
</table>

The chosen colors are considered as points in a color space and the colors of the scale are interpolated on a line through color space connecting these two points. The combo box to the right of the text input field allows using either the HSV or the RGB color space. HSV is fancier and colorful, but RGB is often more useful and readable. For monochrome printing, it is advisable to choose a very light and a very dark color. Choosing white for the minimum and black for the maximum is not at all bad.

By checking the box Manual Scaling it is possible to specify the minimum and maximum values for the color scale in the two text input fields below.

3.2.10.3. Data Tab

The Data tab enables you to choose how the data is analyzed. The Data tab is divided into the Grouping section, the Datum section, the Row Statistics and the Column Statistics.
<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
</table>
| Grouping    | Choose the row and column headers, in other words choose how the data is grouped into categories.  
Choose the groupings for rows and columns independently.  
Note that not all combinations are possible. You cannot have the same header for row and column.  
For example, a matrix cannot be plotted with both the row and column header being Communicator.  
All the headers are explained below. |
| Communicator| Categorize the messages by the MPI communicator.  
The labels are either communicator ids or names. Names are displayed if they are available in the trace file and if they are chosen in the Preferences tab of the Settings dialog box. |
| Collective Operation | Display the types of operations like MPI_Allreduce and MPI_Bcast. |
| Root        | Display the root used in the operation, if applicable. If there is no root, a label No Root is created. |
| Process     | Categorize the operations by the processes. |
| Datum       | Choose which attribute should be printed or painted in the cells. |
| Total Time  | The total time spent in operations, accumulated over all operations and all processes referred in this cell.  
For a single process and a single operation this is the time spent in the call to the operation. For cells referring to a process group this is the sum of the times all contained processes did spent in the operation. For many operations it is the sum over the times spent in each single operation.  
The unit can either be seconds [s] or ticks [tick] depending on the View setting. |
<p>| Minimum Time| The minimum time spent in an operation, minimized over all operations and all processes that fall into this cell ([s] or [tick]) |
| Maximum Time| The maximum time spent in an operation, maximized over all operations and all processes that fall into this cell ([s] or [tick]) |
| Total Volume Sent | The total data volume that has been sent from all operations in this cell [bytes] |
| Minimum Volume Sent | The minimum amount of data volume that has been sent by an operation in this cell [bytes] |
| Maximum Volume Sent | The maximum amount of data volume that has been sent by an operation in this cell [bytes] |</p>
<table>
<thead>
<tr>
<th><strong>Total Volume Received</strong></th>
<th>The total data volume that has been received by all operations in this cell [bytes]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Minimum Volume Received</strong></td>
<td>The minimum amount of data volume that has been sent by an operation in this cell [bytes]</td>
</tr>
<tr>
<td><strong>Maximum Volume Received</strong></td>
<td>The maximum amount of data volume that has been received by an operation in this cell [bytes]</td>
</tr>
<tr>
<td><strong>Total Data Volume</strong></td>
<td>The total data volume, accumulated over all operations in this cell</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>The number of operations in this cell</td>
</tr>
<tr>
<td><strong>Row Statistics</strong></td>
<td>Specify whether the statistical values like the sum, the mean or the standard deviation should be displayed for the rows. Similarly, Column Statistics give the above mentioned statistical values for the given columns.</td>
</tr>
</tbody>
</table>

### 3.3. Font Settings

To change the fonts in any part of Intel Trace Analyzer, open the **Fonts** dialog box from **Options > Set Fonts**.

In the **Fonts** dialog box, you can change the fonts of the labels, legends or headers in the different Charts, for instance. To do it, click the **Change** button and change the size, style and font in the opened dialog box.
3.4. Number Formatting Settings

Use **Number Formatting Settings** to change the way numerical values of different units are represented. This dialog enables you to change the number of digits shown and the format, in which the numerical value is shown for each unit. The number of digits is either interpreted as digits after the decimal point or as the number of valid (non zero) digits. The exact interpretation is dependent on the fact if the chosen format bases on the plain printf-format `%f` or `%g`. 
At the bottom of the **Number Formatting Settings** dialog box, there is a check box that controls repainting. In the checked state, it ensures that all open Views are repainted upon **Apply** or **OK** to take on the new values. Unchecked, the changes in the format become visible only with the next update. All values can be restored to their default by using the **Default All** button or by using the individual **Default** buttons for each number type.

### 3.5. Project Menu

The **Project** menu enables you to save your working environment. The Intel® Trace Analyzer and Collector does not save your environment automatically. You need to save the environment manually.
For the current project files, you can only store views of opened traces and corresponding charts.

You can perform the following actions from the Project menu:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Project...</td>
<td>Open a previously saved project file</td>
</tr>
<tr>
<td>Save Project</td>
<td>Save a project using the default naming</td>
</tr>
<tr>
<td>Save Project As...</td>
<td>Save a project to a specific folder or project name</td>
</tr>
</tbody>
</table>

### 3.6. Windows Menu

Use the menu options in the Windows menu to arrange the open sub-windows as required. The Windows sub-menu also shows the name and path of the trace file that is presently opened.

There are three possibilities to use the Windows menu:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascade</td>
<td>Arrange the open sub-windows one behind the other</td>
</tr>
<tr>
<td>Tile</td>
<td>Arrange the sub-windows next to each other</td>
</tr>
<tr>
<td>Iconify</td>
<td>Minimize the open sub-windows and show their icons in the bottom of the View</td>
</tr>
</tbody>
</table>
NOTE
The **Tile** option is useful only when charts are opened in two or more sub-windows.

### 3.7. Help Menu

The **Help** option enables you to access the HTML formatted help through the integrated HTML browser.

<table>
<thead>
<tr>
<th>Select:</th>
<th>Open:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Trace Analyzer Help</td>
<td>Intel® Trace Analyzer Reference Guide</td>
</tr>
<tr>
<td>Getting Started Guide</td>
<td>Getting Started Guide that describes the initial steps to start using the Intel® Trace Analyzer and Collector</td>
</tr>
<tr>
<td>Tutorial: Detecting and Removing Serialization</td>
<td>Tutorial that helps you learn how to detect and remove unnecessary serialization in your application using the Intel® Trace Analyzer sample trace files</td>
</tr>
<tr>
<td>Intel® Trace Analyzer and Collector Online</td>
<td>Intel® Trace Analyzer and Collector support web page</td>
</tr>
<tr>
<td>Intel® Premier Customer Support</td>
<td>Intel® Premier Customer Support web site where you can submit technical support issues and monitor the previously submitted issues</td>
</tr>
<tr>
<td>About Intel® Trace Analyzer</td>
<td>Version and copyright information for the Intel® Trace Analyzer in use</td>
</tr>
</tbody>
</table>

To ease printing, the *Intel® Trace Analyzer Reference Guide* is also provided in the PDF format.
4. Views

For a flexible analysis of a tracefile, look at multiple partitions of the data from various perspectives using several charts opened in the same View. A View holds a collection of Charts in a single window. The Charts of the same View use one and the same perspective on the data. This perspective is made up of the following attributes:

- time interval
- process aggregation
- function aggregation
- filters

For a detailed explanation of aggregation and filters, refer to Concepts.

Whenever an attribute in the current perspective is changed for one of the Charts, all other Charts change too. You can use several Views at the same time, as it is a very flexible and variable mechanism for exploring, analyzing and comparing trace data.

4.1. View Main Menu

Use the View menu bar to configure the opened Views. The View menu bar consists of the following menu options:

- View Menu
- Charts Menu
- Navigate Menu
- Advanced Menu
- Layout Menu
- Comparison Menu

4.1.1. View Menu

The View tab contains the options that are carried out on the entire View.
These options are:

<table>
<thead>
<tr>
<th>Select This</th>
<th>Shortcut</th>
<th>To Do This</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open New</td>
<td>Ctrl+Alt+O</td>
<td>Open a new View with the Charts previously selected in the Tracefile preferences tab of the Preferences dialog box.</td>
</tr>
<tr>
<td>Tracefile Info</td>
<td>Ctrl+Alt+I</td>
<td>Get full information about the current trace.</td>
</tr>
<tr>
<td>Clone</td>
<td>Ctrl+Alt+D</td>
<td>Create a one-to-one clone of the current View.</td>
</tr>
<tr>
<td>Save</td>
<td>Ctrl+Alt+S</td>
<td>Save the entire View as a picture.</td>
</tr>
<tr>
<td>Print</td>
<td>Ctrl+Alt+P</td>
<td>Print a copy of the View.</td>
</tr>
<tr>
<td>Redraw</td>
<td>Ctrl+Alt+R</td>
<td>Repaint the entire View.</td>
</tr>
<tr>
<td>Close</td>
<td>Ctrl+Alt+W</td>
<td>Close the current View. If the last View for a trace file is closed, the trace file is closed as well.</td>
</tr>
<tr>
<td>Compare</td>
<td>None</td>
<td>Choose the trace files for comparison and compare them with the help of the Comparison dialog box. The traces may represent two runs or two ranges of the same run. Comparison dialog box displays data from the two trace files.</td>
</tr>
</tbody>
</table>

See Also

Preferences
Comparison of two Trace Files
4.1.2. Charts Menu

The Charts menu enables you to open various Intel® Trace Analyzer Charts and use them for trace analysis.

The following Charts are the available:

<table>
<thead>
<tr>
<th>Select This</th>
<th>Shortcut</th>
<th>To Do This</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Timeline</td>
<td>Ctrl+Alt+E</td>
<td>See the individual process activity.</td>
</tr>
<tr>
<td>Qualitative Timeline</td>
<td>Ctrl+Alt+L</td>
<td>See event attributes as they occur over time.</td>
</tr>
<tr>
<td>Quantitative Timeline</td>
<td>Ctrl+Alt+N</td>
<td>See the parallel behavior of the application.</td>
</tr>
<tr>
<td>Counter Timeline</td>
<td>Ctrl+Alt+U</td>
<td>See the values of all counters that are present in the trace file.</td>
</tr>
<tr>
<td>Function Profile</td>
<td>Ctrl+Alt+F</td>
<td>Open the Function Profile. By default, the Function Profile opens for all newly opened trace files.</td>
</tr>
<tr>
<td>Message Profile</td>
<td>Ctrl+Alt+M</td>
<td>Get statistical information on MPI point to point messages.</td>
</tr>
<tr>
<td>Collective Operations Profile</td>
<td>Ctrl+Alt+C</td>
<td>Get statistical information on MPI collective operations.</td>
</tr>
<tr>
<td>Time Scale</td>
<td>None</td>
<td>See the time scale of the project.</td>
</tr>
</tbody>
</table>

See Also
Event Timeline
Qualitative Timeline
Quantitative Timeline
Counter Timeline
4.1.3. Navigate Menu

All navigation options are also available through keyboard shortcuts. These keyboard shortcuts work only when a timeline has the keyboard focus. Left-click on a timeline to give it keyboard focus. When a black rectangle appears around the timeline, it means you have the keyboard focus on it.

The entries in the Navigate menu are:

<table>
<thead>
<tr>
<th>Select This</th>
<th>Shortcut</th>
<th>To Do This</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Trace Map</td>
<td>Ctrl+Alt+A</td>
<td>Enable/disable the Trace Map</td>
</tr>
<tr>
<td>Back</td>
<td>B</td>
<td>See the current interval from the zoom stack.</td>
</tr>
<tr>
<td>Zoom Out</td>
<td>O</td>
<td>Scale by a factor of 0.5 around the current center and push this new interval onto the zoom stack.</td>
</tr>
<tr>
<td>Reset Zoom</td>
<td>R</td>
<td>Clear the zoom stack and push a time interval corresponding to the complete trace file onto the zoom stack.</td>
</tr>
<tr>
<td>Action</td>
<td>Key(s)</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Zoom Up</td>
<td>U</td>
<td>Maintain the current center and go back to the previous zoom step.</td>
</tr>
<tr>
<td>Left 1/1</td>
<td>Ctrl+Left</td>
<td>Move one whole screen to the left and push the new interval onto the zoom stack.</td>
</tr>
<tr>
<td>Right 1/1</td>
<td>Ctrl+Right</td>
<td>Move one whole screen to the right and push the new interval onto the zoom stack.</td>
</tr>
<tr>
<td>Left 1/2</td>
<td>Left</td>
<td>Move half a screen to the left and push the new interval onto the zoom stack.</td>
</tr>
<tr>
<td>Right 1/2</td>
<td>Right</td>
<td>Move half a screen to the right and push the new interval onto the zoom stack.</td>
</tr>
<tr>
<td>Left 1/4</td>
<td>Shift+Left</td>
<td>Move a quarter of a screen to the left and push the new interval onto the zoom stack.</td>
</tr>
<tr>
<td>Right 1/4</td>
<td>Shift+Right</td>
<td>Move a quarter of a screen to the right and push the new interval onto the zoom stack.</td>
</tr>
<tr>
<td>To Start</td>
<td>Home</td>
<td>Move to the start of the trace file and push the new interval onto the zoom stack.</td>
</tr>
<tr>
<td>To End</td>
<td>End</td>
<td>Move to the end of the trace file and push the new interval onto the zoom stack.</td>
</tr>
<tr>
<td>Time Interval</td>
<td>G</td>
<td>Open the Time Interval Selection dialog box.</td>
</tr>
</tbody>
</table>

**See Also**
- Trace Map
- Zoom Stack
- Time Interval Selection

### 4.1.4. Advanced Menu

Use the Advanced menu to tag, filter or idealize your trace files, check imbalance in your application, aggregate processes and functions, or go to a particular function event in the trace.
See the description of the following Advanced Menu options:

- **Tagging Specific Events**
- **Filtering Visible Events**
- **Simulating Ideal Communication**
- **Checking Application Imbalance**
- **Aggregating Results**
- **Aggregating Functions**

### 4.1.4.1. Tagging Specific Events

You can tag, or highlight, events that satisfy specific conditions.

Go to **Advanced > Tagging** and set the conditions for highlighting in the **Tagging** dialog box.

To remove a previously set tag, select **Advanced > Reset Tagging/Filtering**. This command removes all tags and filters.

**See Also**

**Tagging Dialog Box**

### 4.1.4.2. Filtering Visible Events

You can filter the events that are visible in the views. If you use filtering, you see only those events that satisfy specific condition(s). All other events are suppressed as if they were not written to the trace file.

To set the events for filtering, select **Advanced > Filtering**.
To remove a previously set filter, select Advanced > Reset Tagging/Filtering. This command removes all tags and filters.

See Also
Filtering Dialog Box

4.1.4.3. Simulating Ideal Communication

Use the Idealization capability to understand application imbalance and estimate a potential application speedup from tuning MPI implementation and/or network upgrades.

To simulate the application behavior in the ideal communication environment based on the real trace generated by the Intel® Trace Collector, select Advanced > Idealization.

Idealization assumes the following conditions at trace processing time:

- zero network latency
- infinite network bandwidth
- zero buffer copy time
- infinite buffer size

Idealization assumes concurrency as a limitation factor. For example:

- a message cannot be received before it is sent
- an All-To-All collective operation ends when the last thread starts

See Also
Idealization Dialog Box

4.1.4.4. Checking Application Imbalance

Use the Imbalance Diagram to compare the performance between a real trace and an ideal trace.

To do this:

1. Go to Advanced > Imbalance Diagram
2. Choose the traces for comparison in the dialog box that appears
3. Choose the display mode in the resulting Imbalance Diagram dialog box

See Also
Imbalance Diagram Dialog Box

4.1.4.5. Aggregating Results

Use the Process Aggregation capability to set the process groups to focus on and aggregate the results.

To do this, go to Advanced > Process Aggregation.
Processes that are present in the trace file, but not in the process group are represented in the Other group. By default, the data for the Other group is not visible. To view data for the Other group, select Advanced > Show Process Group 'Other'.

To revert to the default aggregation of All processes, select Advanced > Default Aggregation.

See Also
Process Aggregation

4.1.4.6. Aggregating Functions

Use the Function Aggregation capability to set the subset of functions to focus on and aggregate the functions into Function Groups.

To do this, go to Advanced > Function Aggregation.

Functions that are present in the trace file, but not in the function group are represented in the Other group. By default, the data for the Other group is visible. To hide data for the Other group on the Quantitative chart, select Advanced > Show Function Group 'Other'.

To revert to the default aggregation of Major Function groups, select Advanced > Default Aggregation.

See Also
Function Aggregation

4.1.5. Layout Menu

Generally, every View is split into two sections: one for timelines and the other for Profiles. Use the Layout menu to adjust the position of the Charts or the respective sections on the screen.

The Layout options are enabled when two or more charts are open.

The following entries are available in the Layout menu:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timelines to Top</td>
<td>Move the timelines to the top of the View and the profiles to</td>
</tr>
</tbody>
</table>
Timelines to Bottom
Move the timelines to the bottom of the View and the profiles to the top.

Timelines to Left
This moves the timeline(s) being shown in the View to the left of the screen, and also shifts the profiles being shown to the other side.
In case multiple timelines are open, they are collectively shown one below the other on the chosen side.

Timelines to Right
Place all the timelines to the right of the View. It works the same way as the **Timelines to Left** option.

Toggle Timeline Layout
Change the location of the timelines relative to each other. It is useful only when two or more timelines are open at the same time.
If the timelines are stacked and aligned on top of each other, use this option to present them next to each other (or vice versa). The menu entry switches between the two choices.

Toggle Profile Layout
Change the position of the profiles relative to each other, similar to the **Toggle Timeline Layout** option.

Default Layout
Bring the layout of the View back to the default settings. By default, timelines are stacked at the top of the View and Profiles at the bottom, side by side.

For example, select the **Timelines to Right** option to place the Event Timeline on the right and the Function Profile on the left:
4.1.6. Comparison Menu

The Comparison menu tab is displayed in the Views menu bar only when the Comparison mode is activated. To activate it, go to View > Compare.

The Comparison menu contains the following entries:

- Same Time Scaling
- Synchronize Navigation Keys
- Synchronize Mouse Zoom

See Also
Comparison of two Trace Files

4.2. Toolbar

The toolbar provides you with easy access to the Intel® Trace Analyzer functionality. The toolbar is located in the View window and enables you to quickly access the dialogs that are most often used for trace analysis.

The following functionality is available from the toolbar:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracefile Info</td>
<td>Open the Tracefile Info dialogue box, which contains full information about the opened trace.</td>
</tr>
<tr>
<td>Time Scale</td>
<td>Show Time Scale at the top of all the opened Timelines. You can enable the Time Scale only if there is at least one timeline opened.</td>
</tr>
<tr>
<td>Time Interval Selection</td>
<td>Open the Select Time Interval dialog box and select a new time interval for the whole View. To the right of the Time Interval Selection button you can see the data cell that demonstrates the selected time interval. The data in this cell is represented in the form &lt;start&gt;, &lt;end&gt;: &lt;duration&gt;.</td>
</tr>
<tr>
<td>Time Units</td>
<td>Switch time units from seconds to ticks and vice versa.</td>
</tr>
<tr>
<td>Process Aggregation</td>
<td>Open the Process Aggregation dialog box and select the process group for aggregation or create new process groups in addition to those provided by default. The data cell to the right of the Process Aggregation button shows the currently selected process group. By default, it is All_Processes.</td>
</tr>
<tr>
<td>Function Aggregation</td>
<td>Open the Function Aggregation dialog box and choose a</td>
</tr>
</tbody>
</table>
function group for aggregation.

The data cell to the right of the **Function Aggregation** button shows the currently selected function group. By default, it is Major Function Groups.

### Tagging
Tag the necessary messages and collective operations. Intel Trace Analyzer displays the tagged messages and collective operations in bold.

### Filtering
Open the **Filtering** dialog to filter events, messages and collective operations. After you set the filter, the Trace Analyzer displays only the data that you selected in the **Filtering** dialog box.

### Idealization
Open the **Idealization** dialog box to produces an ideal trace.

### Compare Traces
Open the **Comparison View** and calculate the exact differences and speedups between two runs or between two ranges of the same run.

### Imbalance Diagram
Compare your trace with the idealized one. The Imbalance Diagram helps to understand the application imbalance and estimate a potential application speedup from tuning MPI implementation and/or network upgrades.

### Color Editor
Edit the colors of functions and function groups.

### Preferences
Open the **Preferences** dialog box and set up your working environment.

---

**See Also**

- Time Interval Selection
- Process Aggregation
- Function Aggregation
- Tagging Dialog Box
- Filtering Dialog
- Trace Idealizer Dialog Box
- Comparison of Two Tracefiles
- Imbalance Diagram Dialog Box
- Function Group Color Editor
- Preferences

## 4.3. Trace Map

Trace Map enables you to zoom into the relevant subsets of large trace file Charts. It represents a miniature view of the MPI function activity over time. By default, the MPI function activity is shown in red color.
The highlighted part of the Trace Map represents the part of the trace file currently displayed on the opened Chart. This highlighted area - the view area - is updated automatically when you start viewing another time interval of the opened Chart.

When the time interval of one of the Charts changes, the Trace Map and all opened Charts are updated automatically.

NOTE

Trace Map is not currently available in the Comparison mode.

Use the Trace Map context menu to Print, Save or Hide the Chart. To hide the Trace Map through the View Menu, go to View Menu > Navigation > Show Trace Map.

Navigate a chart with the Trace Map:

<table>
<thead>
<tr>
<th>Do This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a specific area of the Trace Map</td>
<td>Change the time interval and zoom into a specific area of the trace</td>
</tr>
<tr>
<td>Press and hold the left mouse button on the selected area of the Trace Map</td>
<td>Move the highlighted view area to the left or to the right of the Trace Map and see the desired part of the opened Chart</td>
</tr>
<tr>
<td>Use the keyboard shortcut keys</td>
<td>Move, expand or shrink the view area</td>
</tr>
</tbody>
</table>
### 4.4. Status Bar

When moving the mouse cursor over an event in the Event Timeline and Counter Timeline or an entry in the Message Profile and Collective Operations Profile, you can see expanded information in the status bar at the bottom of the View for a few seconds.

### 4.5. Navigation in Time

In addition to the menu entries and shortcut, there are two more ways to move in time in Intel® Trace Analyzer:

1. Use the scroll bar found below the timelines
2. Use the mouse to zoom into a time interval

Similar to the keystrokes, these operations manipulate the zoom stack.

To move the View by a quarter of a screen to a selected direction, click the arrow buttons of the scroll bar. To move the View by one entire screen, click into the bar. Both actions push the new interval onto the zoom stack.

To use the mouse for zooming, left-click the timeline (even clicking into the Time Scale works) and select the region of the new time interval, keeping the left mouse button pressed. On releasing the left mouse button this new interval is pushed onto the zoom stack.

Zooming into the Event Timeline:
4.5.1. Zoom Stack

The zoom stack supports navigation in time by storing previously displayed time intervals. You can restore these time intervals when it is necessary. Keyboard or mouse navigation is quite intuitive without detailed knowledge of the zoom stack. Nevertheless, an explanation is given below for the sake of complete reference.

Consider a trace file spanning the time from 0 to 100 seconds. When the first View is opened the zoom stack looks like this:

\[
0: (0;100)(50;100)
\]

Each stack entry has the syntax \((start;stop)(center;width)\). Zooming in using **Zoom In** (I), magnifies the Chart by a factor of 2 around the current center. In the above example, the center is at 50 seconds. Therefore, zooming in twice using the I shortcut key will result in the following stack. You can see that a stack level is created for each action performed. On pressing **Ctrl+Right** twice, the time frame of the Chart is moved to the right by two window sizes:

**State of the Zoom Stack - Zoomed In Twice:**

\[
3: (55;65)(60;10)
\]
For example, here is the original state of the Zoom Stack:

**State of the Zoom Stack - Moved Two Window Sizes to the Right**

0: (0;100) (50;100)
1: (40;80) (60;40)
2: (50;70) (60;20)
3: (55;65) (60;10)
4: (65;75) (70;10)
5: (75;85) (80;10)

Taking this Zoom Stack as an example, see the differences between **Back (B)**, **Zoom Out (O)** and **Zoom Up (U)** explained below.

**Back (B):** When you press B in the original state, one level of the stack pops out and the previous time frame is displayed:

**State of the Zoom Stack after **Back (B)**

0: (0;100) (50;100)
1: (40;80) (60;40)
2: (50;70) (60;20)
3: (55;65) (60;10)
4: (65;75) (70;10)
5: (75;85) (80;10)
6: (70;90) (80;20)

**Zoom Out (O):** Using O in the original state of the Zoom Stack lowers the magnification (doubles the width) and pushes a new item on the stack. Zooming out using O in the original state results in the stack as shown below:

**State of the Zoom Stack after **Zoom Out (O)**

0: (0;100) (50;100)
1: (40;80) (60;40)
2: (50;70) (60;20)
3: (55;65) (60;10)
4: (65;75) (70;10)
5: (75;85) (80;10)
6: (70;90) (80;20)
Zoom Up (U): The resultant zoom after using U is a little more complicated. Pressing U keeps the current center and uses the magnification from before the latest zoom in (which is the change from level 2 to 3 in the original state) and to clear the stack up to and including the last zoom in found. In this example the current center is 80 and the width from stack level 2, which is 20, is used:

State of the Zoom Stack after Zoom Up (U)

2: (70;90) (80;20)
1: (40;80) (60;40)
0: (0;100) (50;100)

Reset (R): Reset clears the stack and pushes one entry covering the full time range of the trace file.
5. Charts

Charts in Intel® Trace Analyzer are graphical or alphanumerical diagrams that are parameterized with a time interval, a process grouping and a function grouping (see Aggregation) and an optional filter. Together they define the structure in which data is presented and the amount of data to be displayed.

The Charts supported by Intel Trace Analyzer are divided into:

1. Timelines: Event Timeline, Qualitative Timeline, Quantitative Timeline and Counter Timeline.

Charts are grouped into Views. These Views provide ways to choose the time interval, the process grouping and optional filters that all Charts in the View use. For more details on Views, see Views.

While the former show trace data in graphical form over a horizontal axis representing runtime, the latter show statistical data. All these Charts are found under Views Menu > Charts. Opening a file in Intel Trace Analyzer displays a View containing the Function Profile Chart (showing the Flat Profile tab) for the opened file.

The following sections describe each Chart in detail. For each Chart there is a subsection about Mouse Hovering, the context menu, the settings dialog box, the effects of filtering and tagging and the effects of aggregation, if it is applicable.

5.1. Event Timeline

The Event Timeline provides a graphical display of the individual process activities over time. To open the Event Timeline, go to Charts > Event Timeline.
Event Timeline

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal bars</td>
<td>Represent the processes with the functions called in these processes. The bars consist of colored rectangles labeled with the function names.</td>
</tr>
<tr>
<td>Black lines</td>
<td>Indicate messages sent between processes. These lines connect sending and receiving processes.</td>
</tr>
<tr>
<td>Blue lines, forming a grid</td>
<td>Represent collective operations, such as broadcast or reduce operations.</td>
</tr>
<tr>
<td>Status bar at the bottom of the panel</td>
<td>Shows the entire runtime of the trace. If the mouse cursor hovers over the Event Timeline, the status bar shows information on events under the cursor. To get a better view of events and communications between them, use zooming.</td>
</tr>
</tbody>
</table>

See Also

Event Timeline Settings
Navigate Menu
5.1.2. Context Menu

You can access the context menu by right-clicking on the Event Timeline. Some general context menu options are common to all Charts. Event Timeline provides some options that are specific to this timeline.

Specific Event Timeline context menu options:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Details on Function/Message</td>
<td>Access details on a particular function or message in the Event Timeline</td>
</tr>
<tr>
<td>Zoom to Collective</td>
<td>Set the View time interval to the time of the respective collective operation</td>
</tr>
<tr>
<td>Show</td>
<td>Select what elements you want to be displayed: functions, messages or collective operations. By default, all of them are displayed in the Event Timeline.</td>
</tr>
</tbody>
</table>

See Also
Common Chart Features

5.1.3. Filtering and Tagging

In the Event Timeline, you see the tagged functions in a frame and with their function labels in bold. The lines of the tagged messages and collective operations are thicker.

When you filter the Event Timeline for a particular process, you see only the MPI functions of this process, since only they pass the filter and all the other functions and processes are filtered out.
Tagging Functions in Process 8 in the Event Timeline
5.2. Qualitative Timeline

The Qualitative Timeline shows event attributes, like the data volume of messages, as they occur over time. To open the Qualitative Timeline, go to **Charts > Qualitative Timeline**.

The value of the event attribute is plotted along the vertical (y) axis and time is plotted along the horizontal (x) axis. Select the required event type and attributes from the context menu. It helps detect patterns and irregular behavior such as extreme deviations or long-term changes in attribute values.

A vertical line in the Qualitative Timeline either represents:

- a single event (denoted as **Single** in the legend)
- several events grouped together (denoted as **Multiple** in the legend)

For the Poisson example, Qualitative Timeline gives a good impression of the pattern of function events. To see it, do the following:

1. Go to **Context Menu > Events to show** and select **Function Events**
2. Zoom into a bunch of iterations

As a result, you see the staircase pattern indicating that your application is serialized:

**Qualitative Timeline**

For more information on detecting and removing serialization in your application, refer to the Tutorial: Detecting and Removing Unnecessary Serialization.

**See Also**
- Qualitative Timeline Settings
- Level of Detail

### 5.2.2. Context Menu

You can access the context menu by right-clicking on the Qualitative Timeline. Some general context menu options are common to all Charts. Qualitative Timeline provides some options that are specific to this timeline.

Specific Qualitative Timeline context menu options:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event to show</td>
<td>Choose the event type from <strong>Function Events</strong>, <strong>Messages</strong> and <strong>Collective Operations</strong></td>
</tr>
<tr>
<td>Attribute to show</td>
<td>Choose the particular attribute value of the event from</td>
</tr>
</tbody>
</table>
NOTE

Not all attributes are available for all event types.

5.2.3. Filtering and Tagging

Use tagging in the Qualitative Timeline to find specific events that occur infrequently. You can find these events because a grouped multiple event is tagged if at least one of the singular events it represents matches the tagging filter expression.

Tagged items in the Qualitative Timeline are highlighted with red.

For example, to tag all messages sent by P3 and P13, do the following:

1. Open the Tagging dialog box and go to the Messages tab.
2. In the group Messages to be Tagged, select the radio button Custom.
3. Specify whatever needs to be tagged in the given field, for example P3, P13.

The result will look similar to:

Tagging Messages in Processes 3,13 in Qualitative Timeline

When you filter the Qualitative Timeline for a particular event, you see only this event, since only this event passes the filter and all the other events are filtered out.
For example, to filter out all messages except those sent by P3 and P13, use the **Filtering Dialog Box**.

The result looks similar to:

**Filtering Messages in Processes 3,13 in Qualitative Timeline**

---

**See Also**
- Filtering Dialog Box
- Tagging Dialog Box
- Tagging and Filtering

---

### 5.3. Quantitative Timeline

The Quantitative Timeline gives an overview of the parallel behavior of the application. It shows over time how many processes or threads are involved in which function. To open the Quantitative Timeline, go to **Charts > Quantitative Timeline**.

Along the time axis, different functions are presented as vertically stacked color bars. The height of these bars is proportional to the number of processes that are currently within the respective function.
You can access the context menu by right-clicking the Quantitative Timeline. Some general context menu options are common to all Charts. Quantitative Timeline provides some options that are specific to this timeline.

Specific Event Timeline context menu options:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ungroup Group MPI</td>
<td>Ungroup the given function group</td>
</tr>
<tr>
<td>Regroup Group MPI</td>
<td>Regroup previously ungrouped function group</td>
</tr>
<tr>
<td>Hide Group MPI</td>
<td>Conceal the chosen activity</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Move Group MPI</td>
<td>Change the position of different groups.</td>
</tr>
<tr>
<td></td>
<td>The selected group can be moved to the top, to the bottom, upward by one position or downward by one position.</td>
</tr>
</tbody>
</table>

**See Also**

*Common Chart Features*

**5.3.2. Filtering and Tagging**

A mesh pattern indicates tagged items in the Quantitative Timeline.

This is what the Quantitative Timeline for the *poisson_icomm.single.stf* trace file with the tagged `MPI_Finalize` function looks like:

**Tagging the MPI_Finalize Function in the Quantitative Timeline**

![Quantitative Timeline with tagged MPI_Finalize function](image)

Filtering in the Quantitative Timeline works the same way as in any other Chart. The result for filtering `MPI_Finalize` looks like:
5.4. Counter Timeline

The Counter Timeline shows the values of all counters that a given trace file provides. The Intel® Trace Collector records user-defined counters or counters provided by the operating system. See the Intel® Trace Collector Reference Guide to learn more about tracing counters.
### Element Description

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line segments</td>
<td>Connect sample points for counters with the Curve attribute</td>
</tr>
<tr>
<td>Markers</td>
<td>Show sample points with the Valid AT point attribute</td>
</tr>
<tr>
<td>Short unconnected horizontal line segments</td>
<td>Show counter samples with other scopes</td>
</tr>
<tr>
<td>Status bar at the bottom of the panel</td>
<td>If the mouse pointer is near a sample point, then its exact position in time is shown.</td>
</tr>
</tbody>
</table>

**NOTE**

There are no markers anymore because the values shown represent samples that were merged over time. To force that effect the View was made very narrow.

**See Also**

- [Counter Timeline Settings](#)
- [Level of Detail](#)

#### 5.4.1. Context Menu

The Counter Timeline context menu provides the common entries as defined in [Common Chart Features](#).
5.4.2. Filtering and Tagging

The filter mechanism in the Intel® Trace Analyzer does not cover counters. Therefore the Counter Timeline is independent of the View filter settings.

5.5. Function Profile Chart

The Function Profile provides detailed profiling information on the performance data.

The Function Profile consists of four tabs: Flat Profile, Load Balance, Call Tree and Call Graph. These tabs use the same column headers and the same raw data. The default column headers on display are Name, TSelf, TTotal, #Calls, and TSelf/Call. For a detailed explanation of all available columns refer to the Function Profile Settings section.

NOTE

The functions in the columns are not in the alphabetical order. You see them in the order given by the layout of the current process or function group.

Function Profile

Do This: | To Do This:
---|---
Drag column headings | Adjust the column order
Click on a column header | Sort a list in the ascending or descending order. The arrow symbol in the column header indicates whether the entries are arranged in the ascending or descending order.
Click TSelf | See which process spends the most time (or the least time) in a function
Go to Options > Preferences > Number Formatting | Set number formatting: increase or decrease the number of digits in numbers shown.
For more information on number formatting, refer to the
See Also
Function Profile Settings
Number Formatting Settings

5.5.2. Flat Profile

By default, the Flat Profile summarizes all major groups of functions and presents statistics over the processes. The exact contents of these groups depend on the group definitions stored in the trace file or you can define them by yourself. In the file `poisson_icomm.single.stf`, these are only MPI and Application groups.

<table>
<thead>
<tr>
<th>Do This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right-click on <strong>Group MPI</strong> and select <strong>Ungroup MPI</strong> from the context menu</td>
<td>See the distribution of execution time over the individual MPI routines. For example, see <strong>Ungrouped MPI</strong>.</td>
</tr>
<tr>
<td>Right-click on the child of <strong>Group MPI</strong> and select <strong>Regroup MPI</strong> from the context menu. Or go to Function Aggregation (<strong>Advanced &gt; Function Aggregation</strong>, or the toolbar button) and select Major Function Groups</td>
<td>Regroup the children of MPI</td>
</tr>
<tr>
<td>Select the <strong>Children of Group All_Processes</strong> entry from the combo box at the top of the tab. See <strong>Selecting Profiles per Process</strong>.</td>
<td>View the data for the children of each process. The result looks similar to <strong>Showing Children of All_Processes</strong>.</td>
</tr>
<tr>
<td>Press the arrows at the side of each process in the <strong>Children of Group All_Processes</strong> view</td>
<td>Expand and collapse the processes of interest</td>
</tr>
</tbody>
</table>
Ungrouped MPI

Selecting Profiles per Process
5.5.3. Load Balance

The **Load Balance** tab displays the same data as the Flat Profile except that it groups the data by function and not by process. The **Load Balance** tab compares the profiles of the same function across several processes. The top level entries of the tree given in the first column are functions.

In this figure you can see that TSelf for MPI_Allreduce is pretty unbalanced across processes:
## Load Balance for MPI_Allreduce

<table>
<thead>
<tr>
<th>Do This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right-click on <strong>Group MPI</strong> and select <strong>Ungroup MPI</strong> from the context menu.</td>
<td>See the distribution of execution time over the individual MPI routines.</td>
</tr>
<tr>
<td>Right-click on the child of <strong>Group MPI</strong> and select <strong>Regroup MPI</strong> from the context menu.</td>
<td>Regroup the children of MPI.</td>
</tr>
<tr>
<td>Or go to Function Aggregation (Advanced &gt; Function Aggregation, or the toolbar button) and select Major Function Groups.</td>
<td></td>
</tr>
<tr>
<td>Select the <strong>Children of Group All_Processes</strong> entry from the combo box at the top of the tab.</td>
<td>View the data for the children of each process.</td>
</tr>
<tr>
<td>Press the arrows at the side of each process in the <strong>Children of Group All_Processes</strong> view.</td>
<td>Expand and collapse the processes of interest.</td>
</tr>
<tr>
<td>Switch between the list and pie charts by pressing the button in the top right corner of the tab.</td>
<td>Analyze the overall load balance pattern (for TSelf). For example, see Pie Charts in the Load Balance Tab.</td>
</tr>
<tr>
<td>Use two spin buttons above the pie charts.</td>
<td>Control the minimum radius of the pies (left button) and how many pie charts appear in a row (right button).</td>
</tr>
</tbody>
</table>

Analyze the overall load balance pattern (for TSelf). For example, see Pie Charts in the Load Balance Tab. However, there may be a huge number of processes in a relatively confined space.
5.5.4. Call Tree

When the Load Balance and Flat Profile tabs show less detail than you need, use the Call Tree tab to analyze calling dependencies. The Call Tree demonstrates the calling hierarchy.

In the Call Tree, select an entry to focus on. If you change the time interval scrolling or zooming, the focus remains on this entry. The time interval stays selected and visible when possible. If there is no corresponding entry for the new time interval, the Call Tree shows its parent. This feature is useful in large and deeply nested call trees.
5.5.5. Call Graph

The Call Graph tab shows a small part of the call graph for each process or process group: a single node (called central function) with its inbound and outbound edges. Each process entry has three children: the Callers, the central function, and the Callees.

Navigating through the Call Graph, double-click on a caller or callee and press the space bar or the Enter key to select the respective function as the central node.

The time shown for the central function is the same as shown in the Flat Profile tab and the Load Balance tab. The time shown for the callers represents the time spent in the central function when called from the respective function.

If your application uses a function in different contexts (for example, different algorithms use the function), you can observe which algorithm causes a function to consume more or less time. In the Call Graph, you can see which caller is responsible for most of the time spent in MPI: it is the function group Application (and not Forward, Adjoin, cg, or Smoother). If the call tree gets too big to navigate through, use the Call Graph to find places in the code that cause expensive calls.
5.5.5.1. Context Menu

Right-click on the chart to open the context menu. The context menu adjusts itself to suit the selected entry in the chart.

**Function Profile Context Menu**

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **Show All_Processes/xxx** | Show the given profile in a different tab.  
Here *xxx* stands for the Function group name                                            |
| **Ungroup**              | Ungroup the selected group and show the distribution of execution time over individual routines                                           |
| **Regroup**              | Restore the summarized display after ungrouping.  
To restore the summarized display after ungrouping a number of times, open the Function Aggregation dialog box (*Advanced > Function Aggregation*) and select *Major Function Groups*. |
| **Find**                 | Search for a process/function                                                                                                                                 |
| **Export (Flat) Data**   | Save the flat profile data in the text form.  
In the **Export Text** dialog box that appears, specify the filename or choose the file for data storage. Saving includes all data of the flat profile along with the child processes. The default option is to save them as a .txt file. |
| **Chart** sub-menu       | Print, save, clone and move the Chart (for more details, see section [Common Chart Features](#) )                                                  |

**See Also**

[Common Chart Features](#)

### 5.5.5.2. Filtering and Tagging

Tagged entries are shown in bold font. Entries with tagged descendants are shown with underlined names. This helps to see or find the required entry, especially when the tree is large. For more details on tagging and filtering, refer to the [Tagging and Filtering](#) section.
5.5.6. Function Profile Settings

The **Function Profile Settings** dialog box enables you to customize displayed options for all the four tabs of the Function Profile Chart. To access the Preferences dialog box, right click on the chart and select **Function Profile Settings** from the context menu.

5.5.6.1. Preferences Tab

There are four groups of options in the **Preferences** tab. Use this tab to set up the display, time, scale bars and colors.

See Also

Tagging and Filtering
# Preferences Tab of the Function Profile Settings

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Self</td>
<td>Display time spent in the given function, excluding time spent in functions called from it.</td>
</tr>
<tr>
<td>Time Total</td>
<td>Display time spent in the given function, including time spent in functions called from it.</td>
</tr>
<tr>
<td>#Calls</td>
<td>Display the number of calls to this function. If other attributes are non-zero, this attribute can be zero, because the actual calls to the respective function can occur outside the current time interval.</td>
</tr>
<tr>
<td>Time Self per Call</td>
<td>Display <strong>Time Self</strong> averaged over <strong>#Calls</strong></td>
</tr>
<tr>
<td>Time Total per Call</td>
<td>Display <strong>Time Total</strong> averaged over <strong>#Calls</strong></td>
</tr>
</tbody>
</table>
### Charts

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#Processes</strong></td>
<td>Display the number of processes in this function</td>
</tr>
<tr>
<td><strong>Time Self per Process</strong></td>
<td>Display <em>Time Self</em> averaged over <strong>#Processes</strong></td>
</tr>
<tr>
<td><strong>Time Total per Process</strong></td>
<td>Display <em>Time Total</em> averaged over <strong>#Processes</strong></td>
</tr>
<tr>
<td><strong>Function Colors</strong></td>
<td>Open the Function Group Color Editor</td>
</tr>
</tbody>
</table>

Use the given check boxes to switch on and off the displaying of the above attributes either as text or as a bar graph. By default, only four boxes are checked - *Time Self*, *Time Total*, *#Calls*, and *Time Self per Call*.

Use the radio buttons to specify the time format (seconds or ticks) or to specify time as the percentage of the time interval.

The scaling modes are given as radio buttons. They are:

- **Visible Items** scales the bars to the respective maximum of all expanded items.
- **All Items** uses the global maximum of all values, regardless of whether they are expanded or not.
- **Siblings** uses only the maximum of the direct siblings.

All the three scaling modes use values only from the same column.

#### 5.5.6.2. Processes Tab

If you want particular process to be displayed in the Function Profile:

1. Go to the **Processes** tab of the Function Profile Settings and choose the necessary processes
2. Choose the **As selected in Settings** option from the combo box

To select all but one process:

1. Choose the process you do not need
2. Using the **Invert All** option to reverse the selection

**NOTE**

It does not influence the current process group of the View, but only focuses the Function Profile on a subset of all processes.

#### 5.5.6.3. Pies Tab

Switch the individual diagram titles and the global legend on and off in the **Pies** tab.

**See Also**

*Function Group Color Editor*
5.6. Message Profile

The Message Profile Chart categorizes messages by groupings in a matrix and shows the value of several attributes in each cell.

By default the matrix is square with the sending processes as row labels and the receiving processes as column labels. It shows in cell \((i, j)\) the total time spent in transferring messages from sender \(i\) to receiver \(j\).

This chart also includes per row and per column statistics, which give the sum, the average and the standard deviation for the respective row or column.

### Message Profile

![Message Profile Chart]

<table>
<thead>
<tr>
<th>Do This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click on row or column headers</td>
<td>Select rows or columns</td>
</tr>
<tr>
<td>Keep the mouse button pressed and select the respective cells</td>
<td>Select an arbitrary area of the matrix</td>
</tr>
<tr>
<td>Right-click and select <strong>Zoom to Selection</strong> from the context menu</td>
<td>Restrict the display to the selected area</td>
</tr>
<tr>
<td>Hold down the [Ctrl] key and drag the header to the required position</td>
<td>Change the position of the row and column headers</td>
</tr>
<tr>
<td>Use the slider above the matrix</td>
<td>Change cell size</td>
</tr>
<tr>
<td>Right-click and select the <strong>Message</strong></td>
<td>Change the grouping that defines the row and column</td>
</tr>
</tbody>
</table>
Profile Settings | headers of this matrix and therefore the categorization of the data.
Available groupings in addition to Sender and Receiver are for example Tag and Communicator.

Use the Message Profile Settings | Choose the attribute to be shown in the cells.
Apart from the Total Time shown by default there are other time values, transfer rates, volumes, and counts.

Press [+] (or [Ctrl]+[+]) | Increase the number of digits locally by three (or one) digits

Press [-] (or [Ctrl]+[-]) | Decrease the number of digits

Place the mouse over any point in the matrix | See the detailed information for the current position in the View status bar in the form $AttributeValue($RowLabel,$ColumnLabel).
It enables you to get exact attribute values even if the cells are configured to be very small or to show no alphanumerical entries at all.

See Also
Message Profile Settings

5.6.2. Context Menu

You can access the context menu by right-clicking the Message Profile. Some general context menu options are common to all Charts. Message Profile provides some options that are specific to this timeline.

Specific Message Profile context menu options:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
</table>
| Attribute to show, Columns to show and Rows to show | Select attributes and groupings.
These entries are the same as those explained in the Message Profile Settings section. |
| Sort | Sort rows by the values of the column clicked on, or to sort columns by the values in a row clicked on and to switch back to the default order. |
| Use the Zoom to the selection option as shown in the example | See the result.
The zoom feature of the Message Profile relies on storing the row and column labels to be suppressed. It can have surprising effects: if Volume is selected as row grouping and the rows with labels 17 and 19 are hidden, then scrolling into an area containing messages with volume 18 results in these messages being shown. To suppress all messages with certain volumes, use filtering (see Filtering). |
Show All  
Show hidden rows and columns

Hide  
Hide some data

Export Data  
Open an Export Text dialog box to select a file to store textual data in. It includes all data cells that contain at least one message, even if they are currently hidden. It does not contain row or column statistics. For each cell, it stores all available attributes.

Zooming to Selected Area in the Message Profile
See Also
Message Profile Settings
Common Chart Features
Filtering Dialog Box

5.6.3. Filtering and Tagging

Gradient in the cell color shows that the cell is tagged. A cell is tagged as soon as a single tagged message exists in it. For example, you can see that messages in Process 3 (P3) are tagged:
Tagging a Process in the Message Profile

Messages that do not pass a filter are not accounted for and may result in a smaller matrix when it occurs in empty rows and columns.

See Also
Tagging and Filtering

5.6.4. Aggregation
The View thread group influences the labels of the Sender, Receiver and Sender/Receiver groups. The View function group has no influence.

If the View shows the thread group Other, then this results in additional rows and columns for the groupings Sender, Receiver and Sender/Receiver.

5.6.5. Message Profile Settings
The Settings dialog box contains three tabs: Preferences, Colors and Data.

5.6.5.1. Preferences Tab
Use the Preferences tab to configure Message Profile display and layout.
Select This:                             To Do This:

Row Labels                             Switch row headers

Column Labels                          Switch column headers

Scale                                  Switch the colored scale next to the matrix

Grid                                   Switch the black grid shown between cells on/off

Keep Empty Rows/Columns when using Sender/Receiver Groupings

Switch the special feature on/off.

This feature is only relevant for the groupings Sender and Receiver.

When you check the box, all processes are shown. For example, even empty rows and columns will be shown. It keeps the form of the matrix constant and makes it easy to look for patterns in the data.
When you uncheck the box, empty rows and columns are suppressed even in these groupings.

All other groupings suppress empty rows and columns to save screen space regardless of the state of this check box.

**Communicator Names**

See helpful communicator names (if available in the trace file).

It may take a lot of valuable screen space.

**Communicator Ids**

Show only concise communicator ids

**Automatic Cell Sizes**

Adjust cell sizes to make all text readable

**Equal Cell Sizes**

Set equal column widths and enable the check box **Square Cell Sizes**

**Square Cell Sizes**

Make cells square

**Manual Cell Sizes**

Specify the size of the cells in pixels either in the **Cell Size** group at the bottom of the tab or using the slider on top of the matrix.

In this mode, the alphanumerical data in the cells is displayed only if it fits or if it is switched off entirely by un-checking the check box **Text in Cells**.

### 5.6.5.2. Colors Tab

In this tab you can change the default colors in which the messages in the Message Profile are shown.
The Message Profile Setting provide the following options:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Color</td>
<td>Choose the colors for the maximum attribute values</td>
</tr>
<tr>
<td>Minimum Color</td>
<td>Choose the colors for the minimum attribute values</td>
</tr>
<tr>
<td>Text input field</td>
<td>Specify the number of color steps (1-255).</td>
</tr>
<tr>
<td>HSV/RGB combo box</td>
<td>Choose between HSV and RGB color space. HSV is more fancy and colorful, but RGB is often more useful and readable.</td>
</tr>
</tbody>
</table>
For monochrome printing, you will get a better result if you choose a very light and a very dark color. For example, white for the minimum and black for the maximum.

**Manual Scaling**

Specify the minimum and maximum values for the color scale in the two text input fields below.

This is very convenient when you compare two Message Profile Charts that may live in different Views.

### 5.6.5.3. Data Tab

The **Grouping** group provides two combo boxes to choose how the data is grouped into categories. The groupings for rows and columns are chosen independently. Note that not all combinations are possible. You cannot set the same grouping for rows and columns and you cannot have Sender/Receiver at one axis and any one of Sender or Receiver on the other axis.

The available groupings are:
<table>
<thead>
<tr>
<th>Grouping</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender</td>
<td>Categorizes the messages by Sender. The exact labels are defined by the current thread group that is given by the View (see the Views section).</td>
</tr>
<tr>
<td>Receiver</td>
<td>Categorizes the messages by Receiver. The exact labels are defined by the current process group that is given by the View.</td>
</tr>
<tr>
<td>Sender/Receiver</td>
<td>Categorizes the messages by Sender/Receiver pairs. The exact labels are defined by the current process group that is given by the View.</td>
</tr>
<tr>
<td>Tag</td>
<td>Categorizes the messages by the MPI tag assigned to the message by the program at the sender side.</td>
</tr>
<tr>
<td>Communicator</td>
<td>Categorizes the messages by the MPI communicator. The labels are either communicator ids or names. Names are displayed if they are available in the trace file and if they are chosen in the Preferences tab of the Message Profile Settings dialog box.</td>
</tr>
<tr>
<td>Volume</td>
<td>Categorizes the messages by their Volume; for example, size in bytes. Grouping Volume by Receiver shows only messages with a volume of 2000 bytes.</td>
</tr>
<tr>
<td>Sending Function</td>
<td>Categorizes the messages by the function that sends them. Labels are names of MPI functions such as MPI_Irsend. This categorization is not influenced by the current Function Aggregation.</td>
</tr>
<tr>
<td>Receiving Function</td>
<td>Categorize the messages by the function that receives them. Labels are names of MPI functions like MPI_Waitany. This is not influenced by the current Function Aggregation.</td>
</tr>
<tr>
<td>Datum group</td>
<td>Allows choosing which attribute should be printed or painted in the cells. See the description of the attributes.</td>
</tr>
<tr>
<td>Row Statistics</td>
<td>group</td>
</tr>
</tbody>
</table>
### Column Statistics group

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Time</td>
<td>The total travel time of the messages, accumulated over all messages that fall into this cell. The unit is either [s] or [tick] depending on the View setting.</td>
</tr>
<tr>
<td>Minimum Time</td>
<td>The minimum travel time of a message, minimized over all messages that fall into this cell. The unit is either [s] or [tick] depending on the View setting.</td>
</tr>
<tr>
<td>Maximum Time</td>
<td>The maximum travel time of a message, maximized over all messages that fall into this cell. The unit is either [s] or [tick] depending on the View setting.</td>
</tr>
<tr>
<td>Average Transfer Rate, [B/s]</td>
<td>The average transfer rate, averaged over the transfer rates of all messages that fall into this cell. Messages are not weighted; for example, transfer rates of short messages have the same impact as transfer rates of long messages.</td>
</tr>
<tr>
<td>Minimum Transfer Rate, [B/s]</td>
<td>The minimum transfer rate, minimized over all messages that fall into this cell.</td>
</tr>
<tr>
<td>Maximum Transfer Rate, [B/s]</td>
<td>The maximum transfer rate, maximized over all messages that fall into this cell.</td>
</tr>
<tr>
<td>Total Data Volume, [B]</td>
<td>The total data volume, accumulated over all messages that fall into this cell.</td>
</tr>
<tr>
<td>Minimum Data Volume, [B]</td>
<td>The minimum data volume, minimized over all messages that fall into this cell.</td>
</tr>
<tr>
<td>Maximum Data Volume, [B]</td>
<td>The maximum data volume, maximized over all messages that fall into this cell.</td>
</tr>
<tr>
<td>Count, [1]</td>
<td>The number of messages that fall into this cell.</td>
</tr>
</tbody>
</table>
5.7. Collective Operations Profile

The Collective Operations Profile enables you to analyze communication patterns that are done using MPI Collective Operations. Like the Message Profile (see Message Profile), the Collective Operations are also represented in a color-coded matrix format. The default matrix shows the type of the Collective Operation as the row label and the process as the column label.

The precision of the values shown can be adjusted as explained in the Message Profile section.

Place the mouse over any point in the matrix and see the detailed information for the current cell in the View status bar in the form $\text{AttributeValue} (\text{RowLabel}, \text{ColumnLabel})$. It enables you to get exact attribute values even if the cells are configured to be very small or to show no alphanumerical entries at all.
Collective Operations Profile

The context menu in the Collective Operations Profile mainly consists of the following entries:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute to show</td>
<td>Select the attributes to be shown in the Collective Operations Profile</td>
</tr>
<tr>
<td>Columns to show</td>
<td>Select if to display Collective Operations Profile by process, by root or by communicator.</td>
</tr>
<tr>
<td>Rows to show</td>
<td>Choose whether the rows of the profile show the Collective Operation, the communicator or the root values.</td>
</tr>
<tr>
<td>Sort</td>
<td>Sort rows by the values of the column clicked on, or sort columns by the values in a row clicked on and switch back to the default order.</td>
</tr>
<tr>
<td>Zoom to selection</td>
<td>Use this entry to focus on a particular region in the matrix.</td>
</tr>
</tbody>
</table>

See Also
Message Profile

5.7.2. Context Menu

The context menu in the Collective Operations Profile mainly consists of the following entries:
To do this, select the required region with the mouse, right-click the selection and choose the **Zoom to selection** entry from the context menu. See the example.

As a result, only the selected region is displayed.

<table>
<thead>
<tr>
<th>Charts</th>
</tr>
</thead>
</table>

| **Hide** | Hide all selected cells. To select the necessary cells, hold down the left mouse button and move over the required region. It also automatically opens the context menu. |
| --- |

| **Show All** | Display all the hidden cells. It is enabled only if cells have been previously hidden using the **Hide** entry. |
| --- |

| **Export Data** | Open an **Export Text** dialog box and select a file to store textual data in. This includes all data cells that contain at least one message, even if they are currently hidden. For each cell, all available attributes are given. It does not contain row or column statistics. |
| --- |

| **Collective Operations Profile Settings** | Open the **Collective Operations Profile Settings** dialog box |
| --- |

### Zooming to Selection in Collective Operations Profile

#### 5.7.3. Filtering and Tagging

Tagged cells are emphasized by a small additional frame around the cell in the color of the alphanumerical entry in the cell. A cell is tagged as soon as a single tagged message falls into that cell. Here is an example of tagging **MPI_Allreduce** function in Collective Operations Profile:
Tagging in Collective Operations Profile

Filtering in the Collective Operations Profile works the same way as in any other Chart: only the chosen Collective Operations are shown, the rest are filtered out.

See Also
Tagging and Filtering

5.7.4. Collective Operations Profile Settings

Use the Collective Operations Settings to change the settings of the Collective Operations Profile and the Chart display. You can change the colors, the layout and the statistical attributes.

The Collective Operation Profile Settings dialog box is divided into three tabs namely the Preferences tab, the Colors tab and the Data tab.

5.7.4.1. Preferences Tab

Use the Preferences tab to adjust the Display settings and the Layout settings.

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Labels</td>
<td>Switch the display of the row headers on/off</td>
</tr>
<tr>
<td>Column Labels</td>
<td>Switch the display of the column headers on/off</td>
</tr>
<tr>
<td>Scale</td>
<td>Display the colored scale that is seen on the right-hand side of the matrix</td>
</tr>
</tbody>
</table>
### Charts

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grid</strong></td>
<td>Displays/removes the black grid in which the cells are placed.</td>
</tr>
<tr>
<td><strong>Keep Empty Rows/Columns when using Sender/Receiver Groupings</strong></td>
<td>Switches a special feature that is only relevant for the Groupings Sender and Receiver. For these groupings, check the box to indicate that all processes should always be shown, like for example, showing even empty rows and columns. It keeps the form of the matrix constant and makes it easy to look for patterns in the data. Uncheck the box to indicate that empty rows and columns even for these groupings are suppressed. All other groupings suppress empty rows and columns to save screen space regardless of the state of this check box.</td>
</tr>
<tr>
<td><strong>Communicator Names</strong></td>
<td>See helpful communicator names (if available in the trace file) that may take a lot of valuable screen space.</td>
</tr>
<tr>
<td><strong>Communicator Ids</strong></td>
<td>Restrict the display to show only concise communicator ids.</td>
</tr>
<tr>
<td><strong>Automatic Cell Sizes mode, checking Equal Cell Sizes</strong></td>
<td>Make each column equal in width. Select it also to enable the check box <strong>Square Cell Sizes</strong>.</td>
</tr>
<tr>
<td><strong>Square Cell Sizes</strong></td>
<td>Get square cells.</td>
</tr>
<tr>
<td><strong>Manual Cell Sizes</strong></td>
<td>Specify the size of the cells in pixels. This is done either in the <strong>Cell Size</strong> group at the bottom of the tab or by sizing the cells manually with the slider that is available on top of the matrix as soon as this setting is applied. In this mode, the alphanumerical data in the cells is displayed only if it fits. Otherwise, it is switched off entirely by unchecking the check box <strong>Text in Cells</strong>.</td>
</tr>
</tbody>
</table>

#### 5.7.4.2. Colors Tab

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximum Color</strong></td>
<td>Choose the colors for the maximum attribute values</td>
</tr>
<tr>
<td><strong>Minimum Color</strong></td>
<td>Choose the colors for the minimum attribute values</td>
</tr>
<tr>
<td><strong>Text input field</strong></td>
<td>Specify the number of color steps (1-255)</td>
</tr>
</tbody>
</table>

The chosen colors are considered as points in a color space and the colors of the scale are interpolated on a line through color space connecting these two points. The combo box to the right of the text input field allows using either the HSV or the RGB color space. HSV is fancier and colorful, but RGB is often more useful and readable. For monochrome printing, it is advisable to choose a very light and a very dark color. Choosing white for the minimum and black for the maximum is not at all bad.

By checking the box **Manual Scaling** it is possible to specify the minimum and maximum values for the color scale in the two text input fields below.
5.7.4.3. Data Tab

The **Data** tab enables you to choose how the data is analyzed. The **Data** tab is divided into the **Grouping** section, the **Datum** section, the **Row Statistics** and the **Column Statistics**.

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grouping</strong></td>
<td><em>Choose the row and column headers, in other words choose how the data is grouped into categories.</em></td>
</tr>
<tr>
<td></td>
<td><em>Choose the groupings for rows and columns independently.</em></td>
</tr>
<tr>
<td></td>
<td><em>Note that not all combinations are possible. You cannot have the same header for row and column.</em></td>
</tr>
<tr>
<td></td>
<td><em>For example, a matrix cannot be plotted with both the row and column header being Communicator.</em></td>
</tr>
<tr>
<td></td>
<td><em>All the headers are explained below.</em></td>
</tr>
<tr>
<td><strong>Communicator</strong></td>
<td><em>Categorize the messages by the MPI communicator.</em></td>
</tr>
<tr>
<td></td>
<td><em>The labels are either communicator ids or names. Names are displayed if they are available in the trace file and if they are chosen in the Preferences tab of the Settings dialog box.</em></td>
</tr>
<tr>
<td><strong>Collective Operation</strong></td>
<td><em>Display the types of operations like MPI_Allreduce and MPI_Bcast.</em></td>
</tr>
<tr>
<td><strong>Root</strong></td>
<td><em>Display the root used in the operation, if applicable. If there is no root, a label No Root is created.</em></td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td><em>Categorize the operations by the processes.</em></td>
</tr>
<tr>
<td><strong>Datum</strong></td>
<td><em>Choose which attribute should be printed or painted in the cells.</em></td>
</tr>
<tr>
<td><strong>Total Time</strong></td>
<td><em>The total time spent in operations, accumulated over all operations and all processes referred in this cell.</em></td>
</tr>
<tr>
<td></td>
<td><em>For a single process and a single operation this is the time spent in the call to the operation. For cells referring to a process group this is the sum of the times all contained processes did spent in the operation. For many operations it is the sum over the times spent in each single operation.</em></td>
</tr>
<tr>
<td></td>
<td><em>The unit can either be seconds [s] or ticks [tick] depending on the View setting.</em></td>
</tr>
<tr>
<td><strong>Minimum Time</strong></td>
<td><em>The minimum time spent in an operation, minimized over all operations and all processes that fall into this cell ([s] or [tick])</em></td>
</tr>
<tr>
<td><strong>Maximum Time</strong></td>
<td><em>The maximum time spent in an operation, maximized over all operations and all processes that fall into this cell ([s] or [tick])</em></td>
</tr>
<tr>
<td><strong>Total Volume Sent</strong></td>
<td><em>The total data volume that has been sent from all operations in this cell [bytes]</em></td>
</tr>
<tr>
<td><strong>Minimum Volume Sent</strong></td>
<td><em>The minimum amount of data volume that has been sent by an</em></td>
</tr>
</tbody>
</table>
### Charts

<table>
<thead>
<tr>
<th>Operation in this cell [bytes]</th>
<th>Maximum Volume Sent</th>
<th>The maximum amount of data volume that has been sent by an operation in this cell [bytes]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Volume Received</td>
<td>The total data volume that has been received by all operations in this cell [bytes]</td>
<td></td>
</tr>
<tr>
<td>Minimum Volume Received</td>
<td>The minimum amount of data volume that has been sent by an operation in this cell [bytes]</td>
<td></td>
</tr>
<tr>
<td>Maximum Volume Received</td>
<td>The maximum amount of data volume that has been received by an operation in this cell [bytes]</td>
<td></td>
</tr>
<tr>
<td>Total Data Volume</td>
<td>The total data volume, accumulated over all operations in this cell</td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>The number of operations in this cell</td>
<td></td>
</tr>
<tr>
<td>Row Statistics</td>
<td>Specify whether the statistical values like the sum, the mean or the standard deviation should be displayed for the rows. Similarly, Column Statistics give the above mentioned statistical values for the given columns.</td>
<td></td>
</tr>
</tbody>
</table>

### 5.8. Common Chart Features

All Charts share some common features that are available through common context menu entries.

<table>
<thead>
<tr>
<th>Select This</th>
<th>Shortcut</th>
<th>To Do This</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print Chart</td>
<td>Ctrl+Shift+P</td>
<td>Print a hard copy of the Chart</td>
</tr>
<tr>
<td>Save Chart</td>
<td>Ctrl+Shift+S</td>
<td>Save the chart as a picture</td>
</tr>
<tr>
<td>Clone Chart</td>
<td>Ctrl+Shift+D</td>
<td>Open exactly the same Chart within the present View. In other words, clone the current Chart.</td>
</tr>
<tr>
<td>Clone Chart in New</td>
<td>Ctrl+Shift+V</td>
<td>Create a new View and open a copy of the</td>
</tr>
<tr>
<td><strong>View</strong></td>
<td>current Chart in the new View.</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Chart to Top/Left</strong></td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Useful when two or more Charts of the same kind (Timelines or Profiles) are open.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the Charts are placed one on top of each other, use this entry to move the selected Chart to the top. If they are placed next to each other, then choose this option to move the select Chart to the left.</td>
<td></td>
</tr>
<tr>
<td><strong>Chart to Bottom/Right</strong></td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Useful when two or more Charts of the same kind (Timelines or Profiles) are open.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the Charts are placed one on top of each other, use this entry to move the selected Chart to the bottom. If they are placed next to each other, choose this option to move the selected Chart to the right.</td>
<td></td>
</tr>
<tr>
<td><strong>Close Chart</strong></td>
<td>Ctrl+Shift+K</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Close the Chart</td>
<td></td>
</tr>
<tr>
<td><strong>Ungroup Function Group</strong></td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ungroup the selected function group</td>
<td></td>
</tr>
<tr>
<td><strong>Regroup Function Group</strong></td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Restore the summarized display after ungrouping</td>
<td></td>
</tr>
</tbody>
</table>
6. Dialogs

Apart from the Settings dialog boxes of each chart, there are a number of other dialog boxes in the Intel® Trace Analyzer. All dialog boxes have the same semantics regarding the buttons OK, Cancel and Apply.

In case the current settings of the dialog boxes are inconsistent or out of bounds, the OK and Apply buttons are both disabled.

Learn more about the Intel® Trace Analyzer dialog boxes:

- Process Aggregation
- Function Aggregation
- Tagging Dialog Box
- Filtering Dialog Box
- Idealization Dialog Box
- Imbalance Diagram Dialog Box
- Function Group Color Editor
- Trace Merge Dialog Box
- Seek and Jump Dialog Box
- Details Dialog Box
- Source View Dialog Box
- Time Interval Selection
- Configuration Dialog Boxes
- Find Dialog Box

6.1. Process Aggregation

Use the Process Aggregation Dialog Box to:

- Select a process group (or function group in general) for process aggregation
- Create new groups beyond the ones that are provided by default

Open the Process Aggregation dialog box through Advanced > Process Aggregation or using the toolbar button.

Group definitions are stored in the .itarc file in your home directory (for an English Microsoft* Windows XP* installation, this is similar to \Documents and Settings\username\.itarc). See Configuration Dialogs for other ways to save, edit and load configuration information.
Process Aggregation

To select a process group for aggregation, select the group by using the mouse or the cursor/arrow keys and press **Apply** or **OK**.

**NOTE**

The dialog box only accepts a single, non-empty process group that contains each of its functions not more than once.

Right-click an item in the editor to bring up a context menu with several entries:
### Context Menu

![Process Aggregation](image)

- **Go to:**
  - **Shortcut**
  - **To Do:**
    - **Undo**
      - **Shortcut**: Ctrl+Z
      - **To Do:** Revert the last actions (delete, cut, move, copy and others)
    - **New Group**
      - **Shortcut**: None
      - **To Do:** Creates a new group as a child of the clicked item
    - **Delete**
      - **Shortcut**: None
      - **To Do:** Remove an item.
        - This entry is disabled for items that originate of the trace file. Only user-created items are deleted.
    - **Rename**
      - **Shortcut**: F2
      - **To Do:** Rename user-created items
    - **Find**
      - **Shortcut**: Ctrl+F
      - **To Do:** Open the **Find** dialog box
    - **Find Next**
      - **Shortcut**: F3
      - **To Do:** Search for the next match if search was started before
    - **Move Item(s)**
      - **Shortcut**: None
      - **To Do:** Move several selected siblings up or down relative to their unselected siblings through a submenu.
        - It is much more convenient to use the keyboard shortcuts shown in the submenu instead of the menu itself.
Select

None

Open a submenu to select subsets of the tree under the clicked item. The entries provided distinguish between selecting processes, selecting process groups or both, and if the selection should include only the direct children or all descendants.

Another way of editing is to drag a group, a process, or the entire current selection and to drop it into a target group. This can result in groups that contain the same function twice or more. In the group hierarchy, such groups are accepted for storage but not for aggregation. Empty groups are deleted automatically when the dialog box is closed.

See Also
Aggregation
Configuration Dialogs
Find Dialog

6.1.2. Comparison Mode

In comparison mode (see Comparison of two Trace Files) the dialog also contains a label and a combo box that enables you to apply the chosen aggregation to the other trace file shown in the View.

Use the If match found option to find a process group of the same name in the other file and select it.

6.2. Function Aggregation

In most respects the Function Aggregation works exactly like the Process Aggregation. It enables you to edit function group definitions and to choose a function group for aggregation in a View. The only addition is that you can assign colors to functions and function groups through the context menu.

Open the Function Aggregation dialog box through Advanced > Function Aggregation or using the toolbar button.

To see the full function name, hover your mouse over a group entry and see a tooltip. The full function name (or static path) reflects the original definition of the function or group as stored in the trace file.

You can also manage function groups through the context menu of the Function Aggregation dialog box.
For example, to change the color of any of the functions, select the **Color** option from the context menu and apply a new color to the given function. To assign the color of the parent node to all the children, use the **Assign Color to Children** entry of the context menu. This entry is enabled only if a parent node is selected.

**See Also**

- Process Aggregation
- Function Group Color Editor

### 6.2.2. Comparison Mode

In comparison mode (see [Comparison of two Trace Files](#)) the dialog also contains a combo box that enables you to apply the chosen aggregation to the other trace file shown in the View.

Use the **If match found** option to find a matching function group according to the matching rules explained in [Mapping of Functions](#) and choose it. It works well for predefined groups. If no match is found, the aggregation for the other file remains unchanged.
Use the **Always (Create matching FGroup)** option to find a match. If a match is not found, Intel® Trace Analyzer creates a matching function group in the name space of the other file.

---

**NOTE**

This operation creates a function group that mimics the hierarchical structure of the original group and contains only the functions that are present in both trace files.

---

### 6.3. Filtering Dialog Box

Use the filtering dialog box to specify filter expressions that describe which function events, messages and collective operations are to be filtered out and which are to be shown.

To access the Filtering dialog box, go to **Advanced > Filtering** or use the ⚒️ toolbar button.
In the **Definition of Filter Expression** section of this dialog box, select the mode to define the filter expression:

<table>
<thead>
<tr>
<th>Choose This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using GUI Interface</td>
<td>Generate filter expressions through a graphical interface</td>
</tr>
<tr>
<td>Manually</td>
<td>Type in the filter expression directly</td>
</tr>
</tbody>
</table>

You can build a filter expression using either the GUI interface or the manual mode, either way the resulting expression is parsed upon each change.
If the current expression cannot be converted into a proper filter definition, a red warning indicating the reason is displayed.

If the expression uses filter attributes that require bypassing them in memory trace cache, a warning message is presented that indicates the need to process data directly from the trace file and the additional time required for the analysis.

See Also
Tagging and Filtering

6.3.1. Building Filter Expressions Using Graphical Interface

Use the **Filtering** dialog box to specify filter expressions separately for function events, messages and collective operations under the respective tabs. You can also use the **Processes** tab to restrict the events to a subset of processes.

Some of the text fields in the filter dialog box can take triplets that are of the form `start:stop:incr` or the short form `start:stop` if the increment is one. Such a triplet describes all numbers greater than or equal to start and smaller than or equal to stop and that are of the form `start+incr*n`. The stop value is optional, too. When stop is omitted, all numbers, beginning from start, match.

6.3.1.1. Processes Tab

Use this tab to specify the processes to pass the filter. This tab has effects on all three sub-expressions: Functions, Messages and Collective Operations.
Show Processes

List the desired processes in the text field to see them in the resulting view and filter out all the other processes. The text field can contain a comma-separated list of process ids, process group ids, triplets thereof, unquoted or double-quoted names of threads, processes or process groups. Names given match all equally named processes/groups.
In the **Process Group Selection** dialog box and select one or more processes or process groups.

<table>
<thead>
<tr>
<th>Three-dot-labeled button</th>
<th>Invert</th>
</tr>
</thead>
</table>
| Open the **Process Group Selection** dialog box and select one or more processes or process groups. | Swap the selection.  
For example, if you want all processes except one to pass the filter, select the process to be filtered out and use the **Invert** check box. It is a logical NOT and it is marked with an exclamation mark before the predicate in the filter expression. |

See the resulting filter expression below the filter clauses. To reuse the expression elsewhere, select and copy the expression in the manual mode of the filtering dialog box.

The context menu of the filter expression contains a Copy (Ctrl+C) entry and a Select All (Ctrl+A) entry.

### 6.3.1.2. Functions Tab

In the **Functions** tab, use the **All**, **None** and **Custom** radio buttons to select the mode of filtering.
To enable the filter clause tab, select the **Custom** radio button. The filter clause consists of the following entries:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions</td>
<td>Use this tab to add functions to the filter. The text field can contain a comma-separated list of functions or function group ids, triplets thereof, unquoted or double-quoted names of functions or function groups that describe functions passing the filter. All given names match all equally named</td>
</tr>
</tbody>
</table>
functions/groups.

Press the three-dot-labeled button to open the **Function Group Selection** sub-dialog and choose the function names from the list.

Check the **Add Function Id** check box to write the name and Id of a selected function into the text field. This is useful to resolve ambiguities of function/group names: if only the name is written into the text field, all functions with this name pass the filter; if the name is replaced by an id, only the function with this id passes the filter, not other functions with the same name.

---

**Processes**

Use this tab to add processes to the filter.

The text field can contain a comma-separated list of process ids, process group ids, triplets thereof, unquoted or double-quoted thread, process or process group names that describe processes and threads in the functions passing the filter. All given names match all equally named processes/groups.

Press the three-dot-labeled button to choose processes from a list.

In the **Process Group Selection** sub-dialog, check the **Add Process Id** check box to write the process name and Id into the text field.

To add another filter clause, press the **Add New Clause** button. To remove an existing filter clause tab, use the **Remove Current Clause** button. Clauses are connected by a logical OR, while attributes from the same tab are connected by a logical AND.

### 6.3.1.3. Messages

To enable the filter clause tab in the Messages tab, select the **Custom** radio button. The filter clause tab has the following entries:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Communicator</strong></td>
<td>The text field can contain a comma-separated list of communicator ids, unquoted communicator names or communicator names in double quotes that pass the filter.</td>
</tr>
<tr>
<td></td>
<td>Press the three-dot-labeled button to choose from the list.</td>
</tr>
<tr>
<td><strong>Tag</strong></td>
<td>The text field can contain a comma-separated list of non-negative integers and triplets that describe tag values that pass the filter.</td>
</tr>
<tr>
<td><strong>Processes</strong></td>
<td>Makes a message pass the filter if either the sender or the receiver matches. Analogous to the logical OR of <strong>Sender</strong> and <strong>Receiver</strong>.</td>
</tr>
<tr>
<td><strong>Sender</strong></td>
<td>The text field can contain a comma-separated list of process ids, process group ids, triplets thereof, unquoted or double-quoted names of threads, processes or process groups that describe processes and threads in the message sender that make the message pass the filter. Given names match all equally named processes/groups.</td>
</tr>
<tr>
<td></td>
<td>Press the three-dot-labeled button to choose from the list.</td>
</tr>
</tbody>
</table>
**Table:**

<table>
<thead>
<tr>
<th><strong>Receiver</strong></th>
<th>Analogous to <strong>Sender</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ranks</strong></td>
<td>Makes a message pass if either the sender or the receiver matches. Analogous to the logical OR of <strong>Rank of Sender</strong> and <strong>Rank of Receiver</strong>.</td>
</tr>
<tr>
<td><strong>Ranks of Sender</strong></td>
<td>The text field can contain a comma-separated list of nonnegative integers and triplets that describe sender ranks (in the MPI communicator) that can pass the filter.</td>
</tr>
<tr>
<td><strong>Ranks of Receiver</strong></td>
<td>Analogous to <strong>Rank of Sender</strong>.</td>
</tr>
<tr>
<td><strong>Message Size</strong></td>
<td>The text field can contain a comma-separated list of non-negative integers and triplets that describe message sizes (in bytes) that pass the filter.</td>
</tr>
<tr>
<td><strong>Start Time</strong></td>
<td>The text field can contain a comma-separated list of non-negative integers and triplets that describe start time (in ticks) of the message that makes the operation pass the filter. Press the three-dot-labeled button to enter/edit the time in ticks or seconds (default depending on the View current time unit).</td>
</tr>
<tr>
<td><strong>End Time</strong></td>
<td>Analogous to <strong>Start Time</strong>.</td>
</tr>
<tr>
<td><strong>Duration</strong></td>
<td>The text field can contain a comma-separated list of non-negative integers and triplets that describe the duration (in ticks) of the message that makes the operation pass the filter. Press the three-dot-labeled button to enter/edit the duration (shown as a time interval) in ticks or seconds (default depending on the View current time unit).</td>
</tr>
</tbody>
</table>

**6.3.1.4. Collective Operations**

To enable the filter clause tab in the **Collective Operations** tab, select the **Custom** radio button. The filter clause tab has the following entries:

<table>
<thead>
<tr>
<th><strong>Entry</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Communicator</strong></td>
<td>The text field can contain a comma-separated list of communicator ids, unquoted communicator names or communicator names in double quotes that pass the filter. Press the three-dot-labeled button to choose from the list.</td>
</tr>
<tr>
<td><strong>Collective Operation</strong></td>
<td>The text field can contain a comma-separated list of unquoted or double-quoted names of collective operations that pass the filter. Press the three-dot-labeled button to choose from the list.</td>
</tr>
<tr>
<td><strong>Processes</strong></td>
<td>The text field can contain a comma-separated list of process ids, process group ids, triplets thereof, unquoted or double-quoted names of threads, processes or process groups that pass the filter. Given names match all equally named processes/groups.</td>
</tr>
</tbody>
</table>
6.3.2. Building Filter Expressions Manually

In the Manual mode, construct any filter expression that is valid as described by the expression grammar in the Filter Expression Grammar section.

Use the context menu entries to select processes, functions, communicators and collective operations from a dialog box in the same way as from the graphical interface and to insert them into the expression at the current cursor position.

The percentage sign \% inserts single line comments and there are context menu entries to comment out (or in) selected text blocks.

There is no operator precedence in the Intel® Trace Analyzer; the expressions are evaluated from left to right. However, you can use parentheses if needed.
6.3.3. Filter Expression Grammar

The filter expression grammar creates a filter, which is a set of function \((\text{funcfilter})\), message \((\text{p2pfilter})\), and collective operation \((\text{collfilter})\) filters, each defining a filter for its respective kind of data. These sub-filter specifications are separated by the \# sign and come in any order. Each filter class is specified either once, or more than once (in this case a Boolean
AND is created from all subfilters for a given class), or not specified at all. An example where three classes of filters are specified is the expression generated in the graphical interface.

Each filter class specifier (funcfilter, p2pfilter, or collfilter) is followed by an expression put in parentheses. That expression can consist of any number of predicates that are different for each filter class and correspond to the entries described in the section Building Filter Expressions Using the Graphical Interface. (See Formal Description of the Grammar) These predicates are joined by using Boolean AND (&&) and OR (||) operators. Boolean expressions are parenthesized as needed. Also, a Boolean NOT (!) operator in front of any predicate or parenthesized expression negates the predicate/expression.

A filter class enables you to define a special expression to pass all or no data through the filter. For example, p2pfilter(NONE) filters out all messages, while collfilter(ALL) passes all collective operation. When the keywords ALL or NONE are used, ensure that it is the only argument to funcfilter, p2pfilter, or collfilter.

Keyword specification in the filter expression grammar is case-insensitive. Specifying names (for functions, processes and communicators, among others), however, is case-sensitive. Use double quotes for names that consist of several words or do not start with a letter or an underscore character (for example, "Major Function Groups"). Use double quotes for single word names (for example, "MPI") if necessary. White space (space and tab characters, as well as newlines) is ignored, unless it is part of a quoted name. If a process/group or function/group name is ambiguous, it is evaluated as if all matching groups were given.

6.3.3.1. Formal Description of Grammar

Here is a formal description of the filter expression grammar:

```plaintext
# The filter itself
FILTER ::= AFILTER | FILTER # AFILTER

# Specifying functions
AFILTER ::= funcfilter ( FUNCFILTARG )
     | collfilter ( COLLFILTARG )
     | p2pfilter ( P2PFILTARG )
# Specifying functions
FUNCFILTARG ::= FUNCEXPR
     | all
     | none
FUNCEXPR ::= FUNCATOM
     | FUNCEXPR & & FUNCATOM
     | FUNCEXPR | | FUNCATOM
FUNCATOM ::= TG
     | FG
     | STARTTIME
```
# Specifying messages

P2PFILTARG ::= P2PEXPR

| all
| none

P2PEXPR ::= P2PATOM

| P2PEXPR && P2PATOM
| P2PEXPR || P2PATOM

P2PATOM ::= DURATION

| COMM
| TAG
| P2PVOLUME
| TGSENDER
| TGRECEIVER
| COMMSENDER
| COMMRECEIVER
| TGSRPAIR
| COMMSRPAIR
| TG
| COMMSR
| STARTTIME
| ENDTIME
| SENDER_FG
| RECEIVER_FG
| ( P2PEXPR )
| ! P2PATOM

# Specifying collective operations

COLLFILTARG ::= COLLEXPR

| all
| none

COLLEXPR ::= COLLATOM
COLLEXPR && COLLATOM
COLLEXPR || COLLATOM
COLLATOM ::= DURATION
| COMM
| COLLOPTYPE
| COLLVOLUME
| TGROOT
| COMMROOT
| TG
| STARTTIME
| ENDTIME
| ( COLLEXPR )
| ! COLLATOM
# Specifying times
STARTTIME ::= start ( TRIPLETS ; INTEGER )
| start ( TRIPLETS )
ENDTIME ::= end ( TRIPLETS ; INTEGER )
| end ( TRIPLETS )
DURATION ::= duration ( TRIPLETS )
# Specifying TGroups and FGroups
TG ::= tg ( NAMES )
FG ::= fg ( NAMES )
SENDER_FG ::= send_fg ( NAMES )
RECEIVER_FG ::= recv_fg ( NAMES )
# Specifying collective operation and message properties
COMM ::= comm ( TRIPLETS )
TAG ::= tag ( TRIPLETS )
COLLOPTYPE ::= type ( COLLNAMES )
COLLVOLUME ::= volume ( TRIPLETS )
P2PVOLUME ::= volume ( TRIPLETS )
# Specifying root, sender, or receiver, either by a TGroup name
# or by position in the communicator. If the operation has no
# root, then root() is always false.

TGROOT ::= root ( NAMES )

TGSENDER ::= sender ( NAMES )

TGRECEIVER ::= receiver ( NAMES )

# The predicate sr specifies both sender and receiver, separated
# by a semicolon.

TGSRPAIR ::= sr ( NAMES ; NAMES )

COMMSROOT ::= root@ ( TRIPLETS )

COMMSENDER ::= sender@ ( TRIPLETS )

COMMSRECEIVER ::= receiver@ ( TRIPLETS )

# The predicate sr@ specifies both sender and receiver ranks,
# separated by a semicolon.

COMMSRPAIR ::= sr@ ( TRIPLETS ; TRIPLETS )

COMMSR ::= tg@ ( TRIPLETS )

# Names containing fancy characters have to be double-quoted.
# Names map to TGroup and thread names, FGroup and function
# names, or collective operation types, depending on the context.

NAMES ::= NAME

| TRIPLET
| NAMES , NAME
| NAMES , TRIPLET

COLLNAMELIST ::= COLLNAMELIST

| TRIPLETS

COLLNAMELIST ::= NAME

| COLLNAMELIST , NAME

NAME ::= [_a-zA-Z][_a-zA-Z0-9.]*

| "[^"]*"\""# Specifying triplets and numbers

TRIPLETS ::= TRIPLET

| TRIPLETS , TRIPLET

TRIPLET ::= INTEGER

| INTEGER :
6.3.3.2. Examples of Advanced Usage of Grammar

This section includes several examples of the manual usage of the filter expression grammar and how the manual usage of the grammar provides advanced capabilities in filtering trace data and speeding up the process of selecting exactly what you want to analyze.

For the first example, consider a parenthesized structure that cannot be built in the graphical interface easily (messages sent by process 0 and starting or ending between 70000 and 80000 ticks):

\[ p2pfilter( \text{sender( 0 ) && ( start( 70000:80000 ) || end( 70000:80000 ) )} ) \]

The following example uses the \text{sr} predicate, which is not available in the graphical interface. This predicate helps to efficiently filter out all messages between processes 0 and 1:

\[ p2pfilter( ! \text{sr( 0:1; 0:1 )} ) \]

Finally, consider the following scenario. With the graphical interface, a complicated filter is specified for a certain filter class with a large number of predicates and Boolean operators (both AND and OR, the latter added by using the \text{Add New Clause} button). To negate everything that has been specified so far (that is, to get exactly the trace data that was previously being filtered out), use \text{!} in front of the whole expression when in the manual mode. For example, the filter below specifies MPI_Barrier collective operations that last no longer than 2000 ticks, plus all collective operations with process 0 as the root:

\[ \text{collfilter( type( MPI_Barrier ) && duration( 0:2000 ) || root( 0 ) )} \]

while the following filter specifies all the collective operations that do not match the description above:

\[ \text{collfilter( ! ( type( MPI_Barrier ) && duration( 0:2000 ) || root( 0 ) ) )} \]

6.3.4. Filter Expressions in Comparison Mode

In the Comparison mode (see \text{Comparison of two Trace Files}), the Filtering dialog has an additional label and a combo box that enables you to apply the chosen filter expression to the other trace file shown in the View.

In the combo box, choose the \text{If expression is valid in other file} option to check if the resulting expression is valid in the name space of the other file. If so, the filter applies the expression as if you had manually typed it in the other file filter dialog box. Any previous input in the target dialog box is overwritten.
6.4. Tagging Dialog Box

For explanation of the filtering and tagging concept refer to the Tagging and Filtering section.

Configure the Tagging dialog box in the same way as you configure the Filtering dialog box. For description, see Filtering Dialog Box.

See Also
Filtering Dialog Box
Tagging and Filtering
6.5. Idealization Dialog Box

Use the Idealization dialog box to produce an ideal trace.

To access the Idealization dialog box, go to Advanced > Idealization, or press the toolbar button.

![Idealization Dialog Box]

The dialog box contains the following elements:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Displays the path to the project input file to be converted</td>
</tr>
<tr>
<td>Output file name</td>
<td>Contains the name of the ideal trace output file.</td>
</tr>
<tr>
<td></td>
<td>By default, the output file is placed in the same folder as the input file, with the suffix ideal added before the .stf extension.</td>
</tr>
<tr>
<td></td>
<td>Type in, or browse to a different location as needed.</td>
</tr>
<tr>
<td>Trace conversion parameters</td>
<td>Specifies the time range of conversion, based on the original trace file time.</td>
</tr>
<tr>
<td></td>
<td>The time range is specified in seconds, in the start time and end time fields.</td>
</tr>
<tr>
<td></td>
<td>The default values are obtained from the original trace.</td>
</tr>
<tr>
<td>Progress bar</td>
<td>Indicates the percentage of conversion completion.</td>
</tr>
<tr>
<td></td>
<td>Before clicking the Start button, you see the progress label Ready to Start. After you click the Start button, the label changes to Converting.</td>
</tr>
<tr>
<td>Open the new trace in a view</td>
<td>Check this box to open the newly produced trace in a separate Intel® Trace Analyzer view</td>
</tr>
<tr>
<td>Create a single STF file</td>
<td>Check this box to store the output trace as a single file. By default, the new trace is created in a multiple-file format.</td>
</tr>
</tbody>
</table>

### 6.6. Imbalance Diagram Dialog Box

The Imbalance Diagram enables you to compare the performance between a real trace and an ideal trace.

Use the Imbalance Diagram total mode to see the time spent on calculation, interconnect, and imbalance as well as the time distribution between them. The diagram displays the time distribution in different colors.

Use the Imbalance Diagram breakdown mode to see:

1. how three most time-consuming functions of the application work
2. how much time they spend on three different kinds of messages
3. what the proportion between interconnect and imbalance for each kind of messages is

The messages can be:

1. short, 0-512 bytes
2. medium, 512-32767 bytes
3. large, > 32767 bytes

To compare a real trace with an ideal trace, take the following steps:

1. Go to Advanced > Imbalance Diagram or press the toolbar button .
2. In the Choose Idealized Trace dialog box, choose a real and an ideal trace that you want to analyze. For example:
3. You can specify the following Imbalance Diagram parameters:

1. Breakdown message thresholds
2. BDI files rebuilding

If you have changed the BDI files manually or the files were partially corrupted, select the **Rebuild the BDI files** checkbox.

The BDI files are the ASCII files and contain all the information needed to create an Imbalance Diagram View. The file name is `<original trace name>.bdi`. The file format is the following:

```
<version string>

<short message threshold> <medium message threshold>

<Function ID> <Collective ID> <Function Name> <Total Function Time> <Short Messages Time> <Medium Messages Time> <Large Messages Time>
```

4. Click **OK**.

**NOTE**

Do not change the BDI file manually. It may result in incorrect feature functioning. You can import the BDI file into a Microsoft Excel* table.
The **BDI File Building** dialog box displays the progress of BDI files building. To stop the BDI building process, click **Cancel**.

**NOTE**

If you stop the BDI building, the **Imbalance Diagram** dialog box is not displayed.

In the **Imbalance Diagram** dialog box, specify whether to display and the mode to display (Total mode or Breakdown mode) the application time. For example:

**Imbalance Diagram View (Total Mode)***
To change the colors used in the Imbalance Diagram, do the following:

1. In the **Imbalance Diagram** dialog box, click **Colors**.
2. In the **Imbalance Diagram Colors** dialog box, click **Choose Color**.
3. Choose the colors and click **OK** to save the changes.

**NOTE**

The new colors are stored in the Intel Trace Analyzer Resource file, so you don't need to change them every time you start the Intel Trace Analyzer. To restore the default colors for all elements, click **Reset all colors**.

**See Also**

*Idealization Dialog Box*

### 6.7. Function Group Color Editor

The Function Group Color Editor is the Function Aggregation in a restricted mode that only enables you to edit the colors of functions and function groups.

You can access the Function Group Color Editor through the Event Timeline settings dialog box or through the context menu of the Function Aggregation dialog box context menu.

The context menu of the Function Group Color Editor has a **Find** entry and a **Find Next** entry, both of which are explained in *Process Aggregation* and also a **Color** entry and an **Assign Color to Children** entry, which are explained in *Function Aggregation*. 
See Also
Process Aggregation
Function Aggregation
Event Timeline
Event Timeline Settings

6.8. Trace Merge Dialog Box

Use the Trace Merge dialog box to merge a raw trace.
The dialog box contains the following elements:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File</strong></td>
<td>See the path of the filename to be merged</td>
</tr>
<tr>
<td><strong>Output file name</strong></td>
<td>Contains the name of the output file for a merged trace. By default, the output file is placed in the same folder of the input file, with the suffix <code>merged</code> added before the <code>.stf</code> extension. Type in, or browse to a different location as needed.</td>
</tr>
<tr>
<td><strong>Progress bar</strong></td>
<td>See the percentage of merging done. Before clicking the Start button, the progress label is <strong>Ready to Merge</strong>. After clicking the Start button, the label changes to <strong>Initializing</strong>, and then to <strong>Merging</strong>.</td>
</tr>
<tr>
<td><strong>Open the new trace in a view</strong></td>
<td>Check this box to open the newly produced trace in a separate Intel® Trace Analyzer view</td>
</tr>
<tr>
<td><strong>Create a single STF file</strong></td>
<td>Check this box to store the resulted trace as a single file. By default, the new trace is created in the multiple-file format.</td>
</tr>
<tr>
<td><strong>Overwrite file if exists</strong></td>
<td>Check this box if you want to overwrite existing output file</td>
</tr>
<tr>
<td><strong>Delete raw data</strong></td>
<td>Check this box if you want to automatically delete the raw trace after merging</td>
</tr>
<tr>
<td><strong>Create summary data files</strong></td>
<td>Check this box if you want to create summary data files. The summary data files can help reduce the time when you first open a trace file.</td>
</tr>
</tbody>
</table>
6.9. Seek and Jump Dialog Box

The seek and jump function enables you to go directly to a particular function event in the trace.

**NOTE**

Use this function with the indexed tracefile format (ITF).

![Seek and Jump dialog](image)

Use the feature to select a particular function event for different processes. Seek and Jump function automatically detects the total number of processes and the number of each function calls in a process, and updates the dialog accordingly to the detected data.

If you select the function, the **Jump** button becomes enabled. Type the desired call number and press **Jump**. The Intel Trace Analyzer zooms all the charts in the **View** to the selected function event.

6.10. Details Dialog Box

The dialog box shows detailed attributes of the clicked events. Function events, messages and collective operations are shown in separate tabs. Each tab shows a list of event entries.

This dialog box is available from the context menus of the Event Timeline and the Qualitative Timeline.
Details on Messages in the Qualitative Timeline

If the column Count shows a value greater than one, the event was created by merging several atomic events. Each entry representing a merged event shows a View, which focuses on the data that went into this entry using the drill down button shown next to the entry. This is called a Detail View; it is a full blown View without restrictions. Use the check-box below the list to filter out the other event categories in the Detail View to be opened.

**NOTE**

Left-click the Details dialog box to reuse an existing Detail View so that the screen is not cluttered so easily. You can also right-click the drill down button to open a new Detail View.

If source code location information is available for an entry, then a button Show source appears next to the entry. This button opens a Source View dialog box. The source code location is only available up to a certain time interval. If the time interval is set in such a way that there is no enter event for a function, then the Details dialog is not aware of the source code location and consequently there will not be a Show Source button.

When a Details dialog is opened, it preserves all the settings of its View (aggregations, filters, ticks vs. seconds) and the event results for which it is opened. Further changes to the View, like a change in Aggregation, do not affect the dialog contents. When such a change is made to the View, the dialog View label is made bold and an asterisk character * appears in front of it. This is to indicate that the details shown are not updated to match the View/Chart. When the Chart or the View from which the Details Dialog originated are closed, the dialog is also closed.

**See Also**

Level of Detail
Source View Dialog

**6.10.2. Detailed Attributes of Function Events**

This dialog box provides you with detailed information about function events.
Details on Functions in the Event Timeline

Use the following entries of this dialog box:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name of the selected function. If it represents a Function Group, then it is prefixed with the Group name.</td>
</tr>
<tr>
<td>Process Group</td>
<td>Specify the Process, Function or Group in which the selected event occurred</td>
</tr>
<tr>
<td>Duration</td>
<td>Show the time spent in a given Function/Group.</td>
</tr>
<tr>
<td></td>
<td>For coarser resolutions (Count ≥ 1), the value does not reflect the actual time spent in this function but the length of the time interval over which several function events were merged.</td>
</tr>
<tr>
<td>Start Time</td>
<td>Show the time when the event entered the Function/Group.</td>
</tr>
<tr>
<td></td>
<td>For coarser resolutions (Count ≥ 1) the value represents the start of the time interval over which several function events where merged.</td>
</tr>
<tr>
<td>End Time</td>
<td>Show the time when the event left the Function/Group.</td>
</tr>
<tr>
<td></td>
<td>On coarser resolutions (Count ≥ 1) the value represents the end of the time interval over which several function events where merged.</td>
</tr>
<tr>
<td>Total #Calls</td>
<td>Give the total number of function calls. It contains all calls covered by the function in the given time interval, including calls to functions other than the function clicked.</td>
</tr>
</tbody>
</table>
|                  | If the column Total #Calls shows a value greater than one, the event was created by merging several atomic events.
Each entry representing a merged event shows a View, which focuses on the data that went into this entry using the drill down button shown next to the entry. This is called a Detail View; it is a full blown View without restrictions. In the dialog box below the list is a check box that allows you to filter out the other event categories in the Detail View to be opened.

**NOTE**

Sometimes a Function starts before and/or ends after the displayed time interval. In these possible cases, you may see in the Details dialog a less \(<\) character preceding the numeric value listed under the column Start Time, and a greater \(>\) character preceding the numeric values listed under the columns **End Time** and **Duration**. These numeric values are the boundaries of the current zoom interval.

### 6.10.3. Detailed Attributes of Message Events

This dialog box provides you with detailed information about messages.

This dialog box contains the following entries:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sender</strong></td>
<td>The Sender is the Process, Function or Group which sent the message</td>
</tr>
<tr>
<td><strong>Receiver</strong></td>
<td>Process, Function or Group which received the message</td>
</tr>
<tr>
<td><strong>Duration</strong></td>
<td>Specifies the time taken by the merged operation. It is the difference between Send Time and Receive Time.</td>
</tr>
<tr>
<td><strong>Send Time</strong></td>
<td>Specifies the time when the message was sent.</td>
</tr>
<tr>
<td></td>
<td>If more than 1 message is represented (Count ≥ 1), then the first Send Time of any member in the merge is specified.</td>
</tr>
<tr>
<td><strong>Receive Time</strong></td>
<td>This indicates when the message was completely received.</td>
</tr>
<tr>
<td></td>
<td>If more than 1 message is represented (Count ≥ 1), then the last Receive Time of any member in the merge is specified.</td>
</tr>
<tr>
<td><strong>Volume</strong></td>
<td>The total number of bytes sent with selected message(s)</td>
</tr>
<tr>
<td><strong>Rate</strong></td>
<td>Indicates the rate at which the bytes are transferred. It is calculated using Volume/Duration.</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Specifies the number of messages that are merged into the selection.</td>
</tr>
<tr>
<td><strong>Tag</strong></td>
<td>Specifies the MPI tag of the message. If more than one message is merged together, then the tag of the first message is shown.</td>
</tr>
<tr>
<td><strong>Communicator Name</strong></td>
<td>The name of the MPI communicator on which the message(s) was</td>
</tr>
</tbody>
</table>
The plain ID of the MPI communicator on which the message(s) was (were) sent is specified in this attribute.

The name of the MPI function from which the message(s) was (were) sent is given in this attribute.

Specifies the name of the MPI function which received the sent message(s).

### 6.10.4. Detailed Attributes of Collective Operation Events

Each possibly merged collective operation has a header entry which describes the collective operation as a whole. Use the plus button to get a detailed list of the same information per Process/Group. The exact processes or process groups shown depend on the current process aggregation.

This dialog box contains the following entries:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name/Process</td>
<td>On the per-operation row this column lists the name of the selected operation (Mixed if different operation types were merged). On per-process rows it shows the name of the Process/Group.</td>
</tr>
<tr>
<td>Duration</td>
<td>Last Time minus First Time</td>
</tr>
<tr>
<td>First Time</td>
<td>First time, one of the merged operations was entered.</td>
</tr>
<tr>
<td>Last Time</td>
<td>Last time, one of the merged operations was left.</td>
</tr>
<tr>
<td>Volume Sent</td>
<td>Number of bytes sent. It is the sum of all bytes sent on all merged operations for the per-process rows. The per-operation row sums up all per-process rows.</td>
</tr>
<tr>
<td>Volume Received</td>
<td>Number of bytes received. It is the sum of all bytes received on all merged operations for the per-process rows. The per-operation row sums up all per-process rows.</td>
</tr>
<tr>
<td>Count</td>
<td>Number of collective operations that are merged into the selection.</td>
</tr>
<tr>
<td>Root</td>
<td>The root process.</td>
</tr>
<tr>
<td>Communicator Name</td>
<td>The name of the MPI communicator on which the collective operation(s) was (were) executed.</td>
</tr>
<tr>
<td>Communicator ID</td>
<td>The plain ID of the MPI communicator on which the collective operation(s) was (were) executed.</td>
</tr>
</tbody>
</table>
6.11. Source View Dialog

Use this dialog box to see the source code.

Open the Source View dialog box from the **Details** dialog box if source code location information is available in the trace file. Use PCTRACE to enable source code location tracing (see the *Intel® Trace Collector Reference Guide* for details).

The dialog box consists of a text browser in the center and a list box representing the call stack at the bottom:

<table>
<thead>
<tr>
<th>Select This:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text browser</td>
<td>View a source file. The line corresponding to the current stack level is shown highlighted in reverse video.</td>
</tr>
</tbody>
</table>
Call stack

Select from the stack levels that were stored with the source code location information. Selecting an entry from this list switches the text browser to the file and line number matching the stack level.

By the default, Intel® Trace Analyzer searches source code files in the current directory and the directory of the current trace file. If no source files are found, you can manually specify the source file to load in the Open File dialog box that appears. The specified path is added to the directory search path. Use the entry `File Default/UserDefines SCLSearchPaths` in the configuration dialog box to specify additional directories to be searched.

See Also
Details Dialog
Edit Configuration Dialog

## 6.12. Time Interval Selection

Use this dialog box to enter a new time interval for the whole View or to enter or edit a time interval or duration in a filter expression.

To open this dialog box, go to Navigate > Time Interval... (G) or press the button on the toolbar. When you open this dialog box this way, it enables you to enter a new time interval for the whole View. This interval is pushed onto the zoom stack and the View is updated accordingly.

You can also open the Time Interval Selection dialog box from the filter dialog boxes. In this case the dialog box enters or edits a time interval or duration in a filter expression.

### Selecting a Time Interval

Specify the time interval in ticks or seconds. You can enter the interval either by giving the start and stop or by giving the center and width. It is possible to enter a value that is greater than the
maximum value of the trace file's time interval; in this case, this value is automatically reduced to the maximum value (tmax of the trace).

See Also

Zoom Stack
Filtering Dialog Box
Tagging Dialog Box

6.13. Configuration Dialog Boxes

Use these dialog boxes to manipulate the configuration that is usually stored in the .itarc file in your home directory upon program exit. To get the global configuration information, use the File Default option and to get the per file information, press the respective file name.

The global information consists of the recent file list and the list of search paths that find the source code files. The latter is stored in the File Default/UserDefines SCLSearchPaths option and contains a list of directories, separated by semicolon ;. These directories are searched in the given order for source code files to be shown in the Source View dialog box.

The per-file information holds all user-defined process groups, function groups and function group colors.

See Also

Source View Dialog Box

6.13.1. Load Configuration Dialog

Use this dialog box to read file configuration data in the program. You can choose the information to be loaded on a file-per-file basis. You cannot save a configuration in the Intel® Trace Analyzer directly, but you can save portions of the .itarc file using a text editor and saving it then in the usual manner.

Use the Merge check box with the current configuration to choose if the configuration that is present in memory for chosen files should be replaced by or merged with the configuration from the selected file.

6.13.2. Edit Configuration Dialog

Use this dialog box to edit the configuration. For example, you can change values or remove entries from the configuration.

Use this dialog box together with the Load Configuration dialog box and store per-trace file settings together with the trace file to share the definitions of process groups and function groups in a work group.
The entries are shown with check box like handles that are all checked by default.

To remove the entry from current configuration, uncheck it and press **OK**. To edit a value, double-click on the respective entry.

You can drag and drop group definitions and color assignments from one trace file branch to another or even onto the branch **File Default** and make them available for each new trace file. The dialog box enables you to drag and drop every child node in the configuration tree, and thus the configuration manipulations can lead to surprising results. In such cases, start again with a well-known configuration.

Generally, it is better to avoid moving options whose values are files or lists of files. For example, a cache file usually corresponds to a particular trace file, and moving a **UserDefines Cachefile** to another trace file or the default section would not make sense, unless the intent is to reuse that cache file with a different trace. On the other hand, you can drag and drop color definitions from trace to trace or to default (the latter will define that color for all trace files, unless they provide their own definition). Moving group definitions, however, requires more attention, since group ids in one trace might not make sense in another. To solve such issues, you can examine and edit values.

**6.14. Find Dialog Box**

Use the **Find** dialog box to search for a process or a function.

Open this dialog box through the context menu of the Function Profile and in the context menus of the Function Aggregation and the Process Aggregation dialog boxes.
Use the **Search in** drop-down menu, to identify the search parameter: Name, All Columns or Dynamic Path.

Check the **Case Sensitive** box, to make search case-sensitive.

You can optimize the search process selecting the necessary type of expression. The expressions are:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains (RegExp)</td>
<td>This entry searches for a phrase that contains a regular expression. A regular expression (RegExp) provides a way to find patterns within text. Regular expressions are built up from expressions, quantifiers and assertions. The simplest form of an expression is a character, like x or 5. An expression can also be a set of characters. For example, [ABCD] would match an A or a B or a C or a D. You can get more information on the valid regular expressions at <a href="http://qt-project.org/doc/qt-4.8/qregexp.html#details">http://qt-project.org/doc/qt-4.8/qregexp.html#details</a>.</td>
</tr>
<tr>
<td>Exact Match</td>
<td>This entry searches for the exact match of a given character or set of strings.</td>
</tr>
<tr>
<td>Wildcards (ITC)</td>
<td>This option defines a search for any function that matches the given pattern. Syntax and semantics are the same as in the regular expressions used in the Intel® Trace Collector. The wildcard characters in use are *, **, ? and[]. They match any number of characters except for the colon :. Pattern matching is not case-sensitive. The state or function name that the pattern is applied to consists of a class name and the symbol name, separated by a colon. The colon is special and is not matched by the * or ? wildcard. To match it use **.</td>
</tr>
</tbody>
</table>
Examples of valid Wild Card patterns are:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI:*</td>
<td>(all MPI functions)</td>
</tr>
<tr>
<td>**::<em>send</em></td>
<td>(any function that contains &quot;send&quot; inside any class)</td>
</tr>
<tr>
<td>MPI:<em>send</em></td>
<td>(only send functions in MPI)</td>
</tr>
</tbody>
</table>
This topic describes the correctness checking reports (CCRs). Get information on how to use the CCRs in the following topics:

- Appearance
- Caching

### 7.1. Appearance

The Correctness Checking Reports (CCRs) are shown on three charts:

- Event Timeline CCRs
- Qualitative Timeline CCRs
- Detailed Dialog CCRs

On the first two charts you can get a detailed dialog through the context menu. The detailed dialog also contains details on the CCRs.

#### 7.1.1. Event Timeline Correctness Checking Reports

On the Event Timeline chart correctness checking reports (CCRs) are displayed as yellow-bordered circles. The color of each circle depends on the type of the particular report: if it is an error then the color is black, if it is a warning - the color is grey.

**Event Timeline with CCRs of Both Types**
You can turn on/off the display of CCRs through the context menu: right-click the chart, go to the **Show** submenu and check/uncheck the **Issues** item. The display is turned on by default.

**Context Menu in the Event Timeline**

7.1.2. Qualitative Timeline Correctness Checking Reports

On the Qualitative Timeline chart correctness checking reports (CCRs) are displayed as vertical lines representing the frequency of the CCRs occurrence. The height of each line is constant.

**Qualitative Timeline with CCRs**

You can turn the display of CCRs on/off through the context menu: for example, right-click the chart, go to the **Events to show** submenu and select **Issues**. The height of each CCR line depends on the level of the report. The lines representing errors are twice higher than the lines representing warnings.

The default event is Duration of Collective Operations.
Correctness Checking Reports

Submenu “Events to show” in Qualitative Timeline Context Menu

Submenu “Attribute to show” in Qualitative Timeline Context Menu

7.1.3. Detailed Dialog

If you right-click the point where the correctness checking reports (CCRs) are present, you can get information about them in the detailed dialog box. The information contains all fields of the Report Data structure that comes from STF. Each data item may contain five fields that are specifiers for the particular report. You can expand each report item in the Detailed Dialog to get the information from the specifiers.

Details on Report Dialog

There can be the following fields:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>Expand the drop-down menu to see processes in which the issue occurred</td>
</tr>
<tr>
<td>Show Source</td>
<td>Press this button to get the exact line in the code at which the issue occurred</td>
</tr>
<tr>
<td>Time</td>
<td>See the moment of time (in seconds or ticks) at which the issue occurred</td>
</tr>
<tr>
<td>Type</td>
<td>See the type of the issue in this string</td>
</tr>
<tr>
<td>Level</td>
<td>See the level of the issue in this string. It can be warning or error.</td>
</tr>
<tr>
<td>Entry Time</td>
<td>See the vector containing moments of time for every process involved into the issue</td>
</tr>
<tr>
<td>Entry Process</td>
<td>See the vector containing numbers of the processes involved into the issue</td>
</tr>
<tr>
<td>Header</td>
<td>See the vector of strings containing descriptions of the issue for a particular process</td>
</tr>
<tr>
<td>Call</td>
<td>See the vector of strings containing function calls involved into the issue</td>
</tr>
<tr>
<td>Function</td>
<td>See the vector of functions involved into the issue</td>
</tr>
</tbody>
</table>

The values of fields from **Entry Time** to **Function** can be different; the sizes of these vectors are equal to each other.

### 7.2. Caching

When it is necessary, each correctness checking report (CCR) is cached like other events (Function, Messages, Collective Operations, etc). All CCRs are put into each level of the cache. The CCRs are not crowded so each cache level contains the same information. It may lead to extra memory usage but this is unlikely since the total number of CCRs is not expected to be large.
8. Comparison of two Trace Files

Comparison View calculates the exact differences and speedups between two runs or between two ranges of the same run.

To open a Comparison View for two trace files, open the files and choose View > Compare in one of the trace file views. You can also use the toolbar button.

Comparison View dialog box appears.

Right below the View menu bar, there are two toolbars with labels Trace A and Trace B. These labels denote two compared trace files.

NOTE

The Comparison View inherits the time interval, aggregation and filter settings from the compared trace files.

By default, Comparison dialog box consists of three Charts: Event Timelines for Trace A and Trace B and a Comparison Function Profile.

In the a view of a single trace file, all charts have the same time interval, aggregation and filters. It refers to the Comparison View Charts as well. But since you compare two trace files, the Comparison View holds two sets of time interval, aggregation and filters, one for each compared trace file.

To compare timelines, use Comparison Charts. The Charts menu of a Comparison Chart contains comparing variants of the profiles that calculate differences and speedups between the two runs. For explanations, see Comparison Charts.
To control the comparison view, go to **View Menu > Comparison** and configure the settings according to your needs.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Set This Option To:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Same Time Scaling</strong></td>
<td>Set this option to use the same scaling for all timelines. For example, when you look at 2 seconds of file A and 4 seconds of file B the timelines for A are shortened so that they occupy half the width of the timelines for B to provide easier visual comparison. If the time intervals for A and B differ by more than a factor of hundred, this setting is ignored and the timelines are aligned as usually to avoid numeric exceptions and distorted diagrams.</td>
</tr>
<tr>
<td><strong>Synchronize Navigation Keys</strong></td>
<td>This option controls the behavior of the navigation keys in comparison mode. By default, pressing a navigation key affects only the Chart in focus. For example, if the Chart belongs to file A then the time interval changes for file A only. Set this option to configure the navigation keys to affect both of the zoom stacks(for more information, refer to <a href="#">Zoom Stack</a>).</td>
</tr>
<tr>
<td><strong>Synchronize Mouse Zoom</strong></td>
<td>Set this option to configure the zoom function so that it operates on both timelines simultaneously.</td>
</tr>
</tbody>
</table>

**See Also**

- [Comparison Charts](#)
- [Zoom Stack](#)

### 8.2. Mappings in Comparison Views

It is important to compare only comparable things. When comparing two program runs you can try to compare the time that a process A.P0 spent in a function in run A to the time of another process B.P0 in run B with or without caring for the fact that B.P0 did only half of the work because run B used twice as much processes.

It is quite easy to see that depending on the domain decomposition or load balancing that is done in the application the meaningful mapping between the processes of two runs cannot be determined automatically. There might be even no such mapping: imagine comparing a run that did a domain decomposition of a cube into 8 processes 2x2x2 with a run that used a 3x3x3 decomposition.

But functions and function groups can be mapped between the runs just by their fully qualified name. This works as long as the structure of modules, namespaces and classes is not changed dramatically.

It is next to impossible to even enumerate all combinations of parameters that might have changed between two runs. To foresee all these cases in terms of automatically adapting GUI does not look promising.

Based on these considerations the mappings of processes, functions, communicators and message tags between the runs are handled differently:
Communicators are mapped by their Ids. Message tags are mapped literally by their value.

The mapping of processes and process groups is controlled by choosing the process aggregations for both files as outlined in Mapping of Processes.

The mapping between functions and function groups is handled automatically as outlined in Mapping of Functions.

See a suitable Process Group for the comparison between a 2 and a 4 processor run in the Process Aggregation dialog box:

Creating a Suitable Process Group

8.2.2. Mapping of Processes

Assume that run A had A.P0, A.P1 and run B had B.P0, B.P1, B.P2, B.P3 and assume that A.P0 did the same work as B.P0 and B.P1 and A.P1 did the same work as B.P2 and B.P3. To get a meaningful Comparison Message Profile under these assumptions, choose the aggregation as shown in here and here:
This image demonstrates the comparison of Run A (with 2 processes) with Run B (with 4 processes)

The message profile shows the average transfer rate quotient B/A. The Comparison Message Profile (and in fact the whole Comparison View) maps the senders and receivers of the two runs onto each other in the following way: child number i of run A's process aggregation is always mapped to (compared with) child number i of run B's process aggregation.

8.2.3. Mapping of Functions

Functions of the two trace files A and B are mapped onto each other by their fully qualified names. These names contain not only the mere function name, but a hierarchical name constructed by the Intel® Trace Collector with the use of any information about modules, name spaces and classes that are available at trace time.

For example the fully qualified name of MPI_Allreduce is MPI:MPI_Allreduce because the Intel Trace Collector puts all MPI functions into the group MPI. Function groups defined by the user in the Intel Trace Analyzer have no influence on these full function names. The function aggregation shows the fully qualified name of a function in a small tooltip window when the mouse hovers over an entry.

The mapping of function groups is a little more subtle. For function groups that are within the hierarchy of the automatically created function group Major Function Groups in file A it is tried to find a matching group in B with the same name and nesting level in the corresponding hierarchy in B.
It works quite well for automatically generated groups. For example MPI is always mapped to MPI even if the groups differ because the two program runs did use a different subset of MPI calls. The same is true for groups that were created by instrumentation using the API provided by the Intel® Trace Collector.

When you create new function groups either by using the Function Aggregation dialog box or the ubiquitous context menu entries to ungroup existing function groups for one file then there will be created matching groups for the other file. You can find these read only groups under the header **Generated Groups** in the Function Aggregation dialog box.

**See Also**
*Function Aggregation*

### 8.3. Comparison Charts

This topic describes how to use and configure the following charts:

- **Comparison Function Profile**
- **Comparison Message Profile**
- **Comparison Collective Operations Profile**

#### 8.3.1. Comparison Function Profile

The Comparison Function Profile is very similar to the regular Function Profile. But it does not have the pie diagrams in the **Load Balance** tab and there is no **Call Graph** tab at all.

**Comparison Function Profile Chart with Ungrouped MPI**
The column headers are the same as in the regular Function Profile with the exception of the first column. In the Comparison Function Profile this column header displays the currently selected comparison operation. To change a comparison operation, right click on the Chart and choose Comparison Operation.

Wherever the regular Function Profile shows the name of a function, function group, process, or process group, the Comparison Function Profile shows either a pair of mapped names in the form of "NameA; NameB" if the names are different, "A unmapped; NameB" or vice versa if there is no mapping for one file, or "Name" if the names are equal for both files.

For example, MPI_Sendrecv in A was replaced by MPI_Isend, MPI_Recv and MPI_Waitall in B. The first column of the profile shows which functions are present in which trace; the other columns indicate which of the traces contain valid data - when there is valid data only for one file, no meaningful comparison can be done.

If you choose Major Function Groups and All Functions for the function aggregation in both runs, all fields show remarks indicating that comparison is impossible because of lacking data - at least as long as the current comparison operation needs data from A and B. If you switch to a comparison operation of either A or B, the respective fields show values even when the function aggregations do not fit. For example, there is data for run A and B but in the rightmost profile in some cases there is nothing to calculate the comparison operation B/A on:

Undefined Fields in the Function Profile

The horizontal bars in the graphical columns work similar to the regular Function Profile except when the comparison operation is set to A/B or B/A. In this case the length of the bars is proportional to the logarithm of the given value. A value of for example 1/5 results in a bar from the middle to the left, while a value of 5 results in a bar of the same length but from the middle to the right. So switching between A/B and B/A will mirror the bars around the middle. Please note that the full column width shows at least the range from 1/2 to 2 so that factors near to 1 do not generate big bars that would invite to misinterpret them as very big changes between the two runs.

The Comparison Function Profile context menu has all the entries of the regular Function Profile context menu (refer to Function Profile Context Menu) with the addition of a possibility to choose the comparison operation:
Comparison Function Profile Context Menu with the Available Comparison Operations

The Comparison Function Profile Settings dialog is the same as the one for the regular Function Profile (refer to Function Profile Settings) with an additional tab that enables you to switch the comparison operation.

See Also
Function Profile Context Menu
Function Profile Settings

8.3.2. Comparison Message Profile

The Comparison Message Profile is very similar to the regular Message Profile. The values shown in the cells are calculated using the currently selected comparison operation shown in the title of the Chart. You can switch the comparison operation from the context menu or from the settings dialog. If the comparison operation makes use of A and B and only one of the values is present for a given cell, this cell is labeled with A or B to indicate for which trace file the data is present. If there is no data at all, the cell is empty.

Some groupings depend on the mapping rules in which they either use processes, process groups or communicators as row or column labels. If differing labels are mapped onto the same column or row, then the label is made up of two lines with the first line holding the label for file A and the second for file B. This can happen for the groupings Sender, Receiver, Sender/Receiver and Communicator. For example, see the Comparison Message Profile with poisson_sendrecv.single.stf as run A and poisson_icomm.single.stf as run B:
There can be cases where a whole row or column does not apply to one of the traces. In these cases the double line label holds \textbf{A missing} or \textbf{B missing}. This can happen for the groupings \textbf{Sender}, \textbf{Receiver} and \textbf{Sender/Receiver} that may have unmappable TGroups in the labels, or for groupings like \textbf{Tag}, \textbf{Communicator} or \textbf{Volume} that directly depend on the data that is present in the trace. You can use the Comparison Message Profile \textbf{Settings} dialog box to suppress such rows and columns, but they are shown by default.

The Comparison Message Profile context menu contains all the entries of the regular Message Profile context menu (refer to \textbf{Message Profile Context Menu}) with the addition of the \textbf{Comparison Operation} entry, from the submenu of which you can select the type of comparison operation.

The Comparison Message Profile \textbf{Settings} dialog box has all the entries of the regular Message Profile Settings (refer to \textbf{Message Profile Settings}). But its \textbf{Preferences} tab has two additional check boxes to suppress rows or columns that only apply to one trace, and the \textbf{Data} tab has an additional combo box to choose the comparison operation.
## Available Comparison Operations

- Attribute to show
- Comparison Operation
- Columns show
- Rows show
- Sort
- Zoom to selection
- Hide
- Show All
- Export Data...
- Chart
- Comparison Message Profile Settings...
- Close Chart

### Comparison Collective Operations Profile

The Comparison Collective Operations Profile is similar to the regular Collective Operations Profile. The values shown in the cells are calculated using the currently selected comparison operation shown in the title of the Chart. The comparison operation can be switched from the context menu or from the settings dialog.

Regarding missing values and unmappable column or row labels the Comparison Collective Operations Profile behaves identically to the Comparison Message Profile (see [Comparison Message Profile](#)).

### See Also

- Message Profile Context Menu
- Message Profile Settings

---

**Comparison of two Trace Files**
Comparison Collective Operations Profile

The Comparison Collective Operations Profile context menu has all the entries of the regular Collective Operations Profile Context Menu (see Collective Operations Profile Context Menu) with the addition of the **Comparison Operation** entry. Use the submenu of this entry to select the type of comparison.

The **Settings** dialog box of the Comparison Collective Operations Profile is similar to the regular Collective Operations Profile Settings (refer to Collective Operations Profile Settings). But you can also use the **Preferences** tab of this Settings dialog box to suppress rows or columns that apply to one trace only, and the **Data** tab to choose the comparison operation.

**See Also**
- Comparison Message Profile
- Collective Operations Profile Context Menu
- Collective Operations Profile Settings
9. Command Line Interface (CLI)

The Command Line Interface (CLI) to the Intel® Trace Analyzer enables you to process trace files without a GUI.

Use the CLI to:

- Compute profiling data automatically
- Generate pre-computed trace caches for trace files

To use the CLI, you must use `--cli` as the first argument to switch off the graphical user interface followed by a trace file name and any other CLI options.

For example, to create the cache for `trace.stf` with default resolution, enter the following command line:

```
traceanalyzer --cli trace.stf -c0 -w
```

A batch file to pre-compute caches might look like this:

```
traceanalyzer --cli poisson_icomm.single.stf -c0 -w

traceanalyzer --cli poisson_sendrecv.single.stf -c0 -w

traceanalyzer --cli vtcounterscopec.single.stf -c0 -w
```

**NOTE**

The CLI is for expert use and can be changed with any version without notice.

The command line interface provides the following options:

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>--messageprofile</td>
<td>Perform message profile analysis.</td>
</tr>
<tr>
<td>--collopprofile</td>
<td>Perform collective operation profile analysis.</td>
</tr>
<tr>
<td>--functionprofile</td>
<td>Perform function profile analysis.</td>
</tr>
<tr>
<td>--starttime=TICKS or -sTICKS</td>
<td>Starting time of the analysis.</td>
</tr>
<tr>
<td>--endtime=TICKS or -eTICKS</td>
<td>Ending time of the analysis.</td>
</tr>
<tr>
<td>--tgroup=ID or -tID</td>
<td>Use this thread aggregation.</td>
</tr>
<tr>
<td>--fgroup=ID or -fID</td>
<td>Use this function aggregation.</td>
</tr>
<tr>
<td>--dump=FILE or -oFILE</td>
<td>The file where to store the analysis results.</td>
</tr>
<tr>
<td><strong>funcformat</strong></td>
<td>A string that contains format switchers specifying how the information about functions are printed; the default value is TFNEIS. Possible format options:</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1. f or F</td>
<td>prints the name of the function group</td>
</tr>
<tr>
<td>2. t or T</td>
<td>prints the name of the thread/process group</td>
</tr>
<tr>
<td>3. g or G</td>
<td>prints the number of processes/threads in the group</td>
</tr>
<tr>
<td>4. E</td>
<td>prints self time in ticks</td>
</tr>
<tr>
<td>5. e</td>
<td>prints self time in seconds</td>
</tr>
<tr>
<td>6. I</td>
<td>prints total time in ticks</td>
</tr>
<tr>
<td>7. i</td>
<td>prints total time in seconds</td>
</tr>
<tr>
<td>8. n or N</td>
<td>prints the number of calls</td>
</tr>
<tr>
<td>9. s or S</td>
<td>prints the source code location (if possible)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>messageformat</strong></th>
<th>A string that contains format switchers specifying how the information about point-to-point messages is printed; the default value is 12DdIiXxAauUn. Possible format options:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>prints if the first member of the message is sender and/or receiver</td>
</tr>
<tr>
<td>2</td>
<td>prints if the second member of the message is sender and/or receiver</td>
</tr>
<tr>
<td>D</td>
<td>prints the summary duration in ticks</td>
</tr>
<tr>
<td>d</td>
<td>prints the summary duration in seconds</td>
</tr>
<tr>
<td>v or V</td>
<td>prints the summary amount of bytes sent</td>
</tr>
<tr>
<td>k or K</td>
<td>prints the minimum amount of bytes sent</td>
</tr>
<tr>
<td>l or L</td>
<td>prints the maximum amount of bytes sent</td>
</tr>
<tr>
<td>U</td>
<td>prints the minimum duration in ticks</td>
</tr>
<tr>
<td>u</td>
<td>prints the minimum duration in seconds</td>
</tr>
</tbody>
</table>
### Command Line Interface

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>X</code></td>
<td>prints the maximum duration in ticks</td>
</tr>
<tr>
<td><code>x</code></td>
<td>prints the maximum duration in seconds</td>
</tr>
<tr>
<td><code>I</code></td>
<td>prints the minimum rate in Bytes/tick</td>
</tr>
<tr>
<td><code>i</code></td>
<td>prints the minimum rate in Bytes/second</td>
</tr>
<tr>
<td><code>A</code></td>
<td>prints the maximum rate in Bytes/tick</td>
</tr>
<tr>
<td><code>a</code></td>
<td>prints the maximum rate in Bytes/second</td>
</tr>
<tr>
<td><code>n</code> or <code>N</code></td>
<td>prints the number of messages</td>
</tr>
</tbody>
</table>

A string that contains format options specifying how the information about collective operations is printed; the default value is `12DdIiXxAauUnvwyzk`.

Possible format options:

- `1` - prints the name of the process group
- `2` - prints the name of the operation
- `D` - prints the summary duration in ticks
- `d` - prints the summary duration in seconds
- `U` - prints the minimum duration in ticks
- `u` - prints the minimum duration in seconds
- `X` - prints the maximum duration in ticks
- `x` - prints the maximum duration in seconds
- `I` - prints the minimum rate in Bytes/tick
- `i` - prints the minimum rate in Bytes/second
- `A` - prints the maximum rate in Bytes/tick
- `a` - prints the maximum rate in Bytes/second
- `v` or `V` - prints the summary amount of bytes sent
- `k` or `K` - prints the minimum amount of bytes sent
- `l` or `L` - prints the maximum amount of bytes sent
- `w` or `W` - prints the summary amount of bytes received
- `y` or `Y` - prints the minimum amount of bytes
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-r[FILE]</code></td>
<td>Read the trace cache from the specified (if provided) or default file.</td>
</tr>
<tr>
<td><code>-w[FILE]</code></td>
<td>If a trace cache has been built, write it to the specified (if provided) or default file.</td>
</tr>
<tr>
<td><code>-cRESOLUTION</code></td>
<td>Build a trace cache with the specified resolution. The resolution is given in clock ticks. Higher values result in smaller (coarser) cache files, 0 (zero) is used as the default resolution.</td>
</tr>
<tr>
<td><code>-FFILTER</code></td>
<td>The filter to use for the analysis, specified as a filter grammar string.</td>
</tr>
<tr>
<td><code>--messagefirst=GROUPING</code></td>
<td>The first grouping in the message profile analysis result (first dimension of matrix).</td>
</tr>
<tr>
<td><code>--messagesecond=GROUPING</code></td>
<td>The second grouping in the message profile analysis result (second dimension of matrix).</td>
</tr>
<tr>
<td><code>--collopfirst=GROUPING</code></td>
<td>The first grouping in the collective operation profile analysis result (first dimension of matrix).</td>
</tr>
<tr>
<td><code>--colloppsecond=GROUPING</code></td>
<td>The second grouping in the collective operation profile analysis result (second dimension of matrix).</td>
</tr>
<tr>
<td><code>--summary</code></td>
<td>Generate the application summary sheet with the format that is described below.</td>
</tr>
<tr>
<td><code>-icpf [options] &lt;tracefile&gt;</code></td>
<td>Process a trace file using the specified simulator at runtime. Use the traceanalyzer <code>-icpf</code> option to process your trace files using specific simulator library. In <code>-icpf [options] &lt;tracefile&gt; -- simulator &lt;simulator library&gt;</code>, the [options] can be: <code>-s &lt;NUM&gt;</code> - processes the trace starting at the time (NUM measured in ticks). <code>-e &lt;NUM&gt;</code> - processes the trace to the end time (NUM measured in ticks).</td>
</tr>
</tbody>
</table>
### Command Line Interface

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-w &lt;NUM&gt;</code></td>
<td>processes the trace based on NUM, 0 for STF, 1 for ASCII, else devnull.</td>
</tr>
<tr>
<td><code>-o &lt;new_name&gt;</code></td>
<td>trace output file name.</td>
</tr>
<tr>
<td><code>-u</code></td>
<td>single file mode. The output file is a single STF.</td>
</tr>
<tr>
<td><code>-h</code></td>
<td>prints this message and exits.</td>
</tr>
</tbody>
</table>

**--ideal [options] <tracefile>**

Produce an ideal trace.

Use the traceanalyzer --ideal option to idealize a trace by Ideal Interconnect Simulator. In `--ideal [options] <tracefile>`, the [options] can be:

- `-s <NUM>` - processes the trace starting at the time (NUM measured in ticks; the default value is 0).
- `-e <NUM>` - processes the trace to the end time (NUM measured in ticks; the default value is the end time of the trace).
- `-w <NUM>` - processes the trace based on NUM, 0 for STF, 1 for ASCII, else devnull (the default value is 0).
- `-o <new_name>` - trace output file name.
- `-u` - single file mode. The output file is a single STF.
- `-sp` - shows percent progress indicator.
- `-q` - quiet mode; turns off all output.
- `-h` - prints this message and exits.

**--breakdowns <real_trace_name> <ideal_trace_name>**

Create intermediate *.bdi files that contain all needed information for the Imbalance Diagram.

**--merge <unmerged_trace_name> [merged_trace_name] [-single] [-delete-raw-data] [-sumdata]**

Merge the raw trace.

- `<merged_trace_name>` - if set this option, then the output trace will have this name; otherwise suffix .merged will be added to the original name.
- `-single` - create a single STF file instead of multiple ones
- `-delete-raw-data` - delete the raw trace after merging
- `-sumdata` - create summary data files while
merging

```
--sumdata <trace_name>
```

Create summary data files from an ordinary trace

The application summary sheet consists of a three-line header:

```
<# processes>:<# processes per node>
,application time>:<MPI time>:<IIS time>
<first message size of middle bucket (2)>:
<first message size of highest bucket (3)>
```

The header is followed by these sets of lines, for each of the top ten functions, sorted by descending total time:

```
<Name of MPI_group>:<# involved processes>
<total time in above func for bucket 1>:<for bucket 2>:<for bucket 3>
<total IIS time in above func for bucket 1>:<for bucket 2>:<for bucket 3>
<count in above func for bucket 1>:<for bucket 2>:<for bucket 3>
<total # bytes in above func for bucket 1>:<for bucket 2>:<for bucket 3>
```

In the application summary sheet, IIS stands for Ideal Interconnect Simulator, which predicts MPI behavior on an ideal interconnect.

You can import the application summary sheet to spreadsheet applications such as Microsoft Excel. Fields are separated by colons. Unknown values are indicated by N/A.
10. Custom Plug-in Framework

The Custom Plug-in Framework (CPF) enables you to write your own simulators using the provided API. For example, Ideal Interconnect Simulator (IIS) is implemented using this framework. Custom developed simulators can interact with trace files from the Intel® Trace Analyzer and can model any condition you can think of.

CPF implementation works with traces that were obtained from applications that pass correctly message checking. To use CPF, install Intel® Professional Edition Compilers on the following systems:

- the system in which you are building your simulator
- the system in which you are running CPF

Learn more about the Custom Plug-in Framework:

- Usage Instructions
- Developing Simulators

10.1. Usage Instructions

Use the CLI option --icpf to process your trace files through a specific simulator library. You can use the following command line:

```
traceanalyzer --cli --icpf [options] <tracefile> --simulator <simulator library>
```

where

- `<tracefile>` - is the name of your trace file.
- `<simulator library>` - is the name of your simulator.

10.2. Developing Simulators

The Intel® Trace Analyzer installation includes examples of a simulator. You can customize the examples of the simulator library to create your own simulator.

The examples are located in: `<trace_analyzer_install_dir>/examples/icpf`

You can see the following files in this directory:

- `h_devsim.cpp` - is the simplest template for developing your own simulator.
- `h_doublesim.cpp` - is an example of simulator library with doubled durations of all MPI events.
- `Makefile` - is a makefile for Linux* gmake.
- `Makefile_win` - is a makefile for Windows* nmake.
10.2.1. Building a Simulator Library

To build a simulator library, do the following steps:

1. Set up your development environment. If you use a non Intel compiler, modify the `Makefile` for Linux OS or `Makefile_win` for Windows OS in the simulator development directory. To modify a `Makefile`, open the `Makefile` with any text editor and edit the following lines to change the compiler:

   Makefile:
   
   `SIMCOMP = icc`

2. To build the examples of simulator libraries, you can use the following commands:
   - to build the simplest library, use `make devsim`.
   - to build the library with doubled durations of all MPI events, use `make doublesim`.
   - to build both libraries, use `make` or `make all`.

10.2.2. Writing your Simulator

Before beginning simulator development, you need to understand the virtual class that is inherited by a simulator.

The `CustomPluginFramework.h` header file connects between the Intel Trace Analyzer source and other tools such as simulators. This file contains the following information:

- data structures that are passed to the API
- public virtual functions that are called by CPF
- private functions that can be called by any simulator help/aid with simulator development
- private members that provide a simulator with access to certain aspects about a particular trace

**NOTE**

If you change the file, you will not be able to plug a simulator into the API.

1. You can create your simulator filling virtual functions in `h_devsim.cpp` (or your `.cpp` file if you have renamed it).

**Examples**

1. In this example, durations of all non-blocking sending MPI events in trace file become zero time.

   ```c
   virtual void ProcessNonBlockingSend( FuncEventInfo * _event_, P2PInfo * _message_ )
   {
   ```
Custom Plug-in Framework

```cpp
_event_->_endTime = _event_->_startTime;
}

2. In this example, durations of all collective operations are doubled.

```cpp
virtual void ProcessCollOp( CollOpInfo * _collop_ )
{
    //Traverse vector of CollOpInfo
    for( U4 k = 0; k < _collop_->_threads->count(); k++ )
    {
        (*(_collop_->_threads))[k]->_endTime += ( (*(_collop_->_threads))[k]->_endTime - (*(_collop_->_threads))[k]->_startTime );
    }
}
```

If you need to add more inline functions for your simulator, add them only to `h_devsim.cpp`
or your `.cpp` file. Do not modify the `h_custompluginframework.h` header file.

---

**NOTE**

When modifying end time, ensure that the end time you change is never less than the start time.

---

**NOTE**

All values of time are stored in a simulator in ticks instead of in seconds. To convert ticks in seconds, use the `clockResolution` coefficient.

---

**NOTE**

When you have trouble with trace, it is suggested that you run the trace through Intel Trace Analyzer and Collector message checker before filling a bug report.

---

### 10.2.3. CPF Simulator API

CPF interacts with the simulators through the custom plug-in framework API. The API provides pre-defined functions that the simulators have to implement.

#### Framework Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Explanation</th>
</tr>
</thead>
</table>

---

163
10.2.3.2. Data Structures

**FuncEventData**

The FuncEventData structure represents any function events that occur in a trace on a specific thread and do not span multiple threads. CPF passes a FuncEventData structure to a simulator whenever it encounters any application event, MPI single rank event, or a non-tracing event (such as idle time on a process). The FuncEventData structure also represents a non-blocking sending because a non-blocking sending occurs only on a specific thread.

The only member of FuncEventData that you can modify using a simulator is the end time of the function event. The other parameters are all constant because modifying any of these values would greatly change the fundamental parts of the trace. The constant parameters can be the thread id, function type, size of message (if it is a non-blocking sending), or start time.

**NOTE**

The end time of this event should never be less than the start time. CPF checks this rule when returned from the simulator.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>_threadID</td>
<td>The thread id where this function event occurred.</td>
</tr>
<tr>
<td>_func</td>
<td>The function id of this event. You can look up the id using mapping provided.</td>
</tr>
<tr>
<td></td>
<td>See <a href="#">Private Functions</a> for details.</td>
</tr>
<tr>
<td>_sizeSent</td>
<td>The size of the message sent by the thread.</td>
</tr>
<tr>
<td></td>
<td>This variable is available when the function event is used as part of the collective operation.</td>
</tr>
<tr>
<td>_sizeRecv</td>
<td>The size of the message received by the thread.</td>
</tr>
<tr>
<td></td>
<td>This variable is available when the function event is used as part of the collective operation.</td>
</tr>
<tr>
<td>_startTime</td>
<td>The start time of this function event.</td>
</tr>
</tbody>
</table>
The end time of this function event.

You can modify this variable, but make sure that it is not less then _startTime.

10.2.3.3. P2PInfo

The P2PInfo data structure represents a point-to-point (P2P) message that occurs in a trace. It is also passed to non-blocking API function to give the simulator access to different aspects of the point-to-point message. The values are slightly different depending on the type of a point-to-point message that this data structure represents. See Private Functions for details.

The following table lists the P2Pinfo variables. All variables, except the end times of the send and receive messages, are constant because modifying any of these values would drastically change the fundamental parts of the trace.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>_sendStartTime</td>
<td>This variable contains the send start time or the time when the message is sent.</td>
</tr>
<tr>
<td>_sendEndTime</td>
<td>This variable is only valid on sendrecv non-blocking sends (only in processnonblockingsend()) or on sync sends.</td>
</tr>
<tr>
<td></td>
<td>This variable can be modified, but should never be less then _sendStartTime.</td>
</tr>
<tr>
<td>_recvStartTime</td>
<td>This variable is only valid on sendrecv, sync sends, waitall/waitsomes.</td>
</tr>
<tr>
<td>_recvEndTime</td>
<td>This variable is valid on all cases, can be modified, but should never be less then _recvStartTime.</td>
</tr>
<tr>
<td>_iRecvStartTime</td>
<td>This variable is only valid when P2P message involves a waitall/waitsome.</td>
</tr>
<tr>
<td></td>
<td>This variable represents the start time of i-receive that occurs before the waitall/waitsome for this thread.</td>
</tr>
<tr>
<td>_backwards</td>
<td>This variable indicates that the message send start time &gt; receive end time.</td>
</tr>
<tr>
<td>_sender</td>
<td>The thread id of the sending thread.</td>
</tr>
<tr>
<td>_receiver</td>
<td>The thread id of the receiving thread.</td>
</tr>
<tr>
<td>_communicator</td>
<td>The MPI communicator.</td>
</tr>
<tr>
<td></td>
<td>The P2P message communication is over; there is a communicator to text mapping provided for the simulator in the API.</td>
</tr>
</tbody>
</table>
10.2.3.4. CollOpInfo

The CollOpInfo data structure represents a collective operation that occurs in the trace. In a trace, there is a collective operation that connects the same event on each thread in the trace to one another. This CollOpInfo data structure provides this grouping to the simulator.

All members of the CollOpInfo data structure are constant and cannot be modified except for the FuncEventData events that represent each thread that is involved in the collective operation. To modify end times of the collective operation on each thread, modify the FuncEventData for each thread in the vector in this collective operation. The same rules apply CollOpInfo for FuncEventData.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>_firstTime</td>
<td>The earliest time that a thread enters a collective operation.</td>
</tr>
<tr>
<td>_communicator</td>
<td>Communicator on which collective operation is executed.</td>
</tr>
<tr>
<td>_collOpId</td>
<td>The function id of the collective operation. You can look up what it is using mapping provided. See Private Functions for details.</td>
</tr>
<tr>
<td>_root</td>
<td>The thread id of the root, valid only if the collective operation is a One-To-All or All-To-One.</td>
</tr>
<tr>
<td>_rootIndex</td>
<td>The root's index into the thread vector.</td>
</tr>
<tr>
<td>_threads</td>
<td>The thread vector that represents the individual function events per thread that make up this collective operation.</td>
</tr>
</tbody>
</table>
10.2.3.5. Public Functions

The following topics list the public API functions and general public functions that CPF calls. When CPF is interacting with the API, it first creates a data structure of type described in Data Structures; then passes a pointer to the simulator’s API function. Any modification to this data structure in the API function automatically reflects changes in CPF. These functions do not return any values.

```c
void setup()
```

Use this function to pass important information/characteristics about the trace from CPF to the simulator. This function takes care of initialization of the private member variables in Private Members. Do not modify this function.

**Syntax**

```c
Void setup (const Vector< U4 >, const Vector< U4 >, const Vector< U4 >, const HashMap< U4, const char* >, const char*, const U8, const U8, const U4)
```

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtual void Initialize ( ) = 0;</td>
<td>Your simulator needs to implement this API function. The function is called before the start of processing a trace file, not to be used as a constructor. This function enables simulator extensions to pre-trace file operations before the executable starts processing the trace file.</td>
</tr>
<tr>
<td>virtual void Finalize ( ) = 0;</td>
<td>Your simulator needs to implement this API function. The function is called at the very end of processing a trace file, not to be used as a destructor. This function allows simulator extensions to have a way of doing post-trace file operations before the executable exits.</td>
</tr>
<tr>
<td>virtual void ProcessFunctionEvent ( FuncEventInfo * <em>event</em> ) = 0;</td>
<td>Your simulator needs to implement this API function. This function processes application events that occur in the trace file. The event object is a regular application event in the trace that is processed from the queues.</td>
</tr>
<tr>
<td>Function Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>`virtual void ProcessNonTracingEvent (</td>
<td>Your simulator needs to implement this API function. This function processes non-tracing events that occur in the trace file. This event object is a non-tracing event in the trace that is to be processed from the queues. A non-tracing event is an empty white space at beginning of trace before each thread, or parts of a trace file which have tracing turned off.</td>
</tr>
<tr>
<td>FuncEventInfo * <em>event</em> ) = 0;`</td>
<td></td>
</tr>
<tr>
<td>`virtual void ProcessMPIFunctionEvent (</td>
<td>Your simulator needs to implement this API function. This function processes a single rank MPI event in trace file. This event object is a single rank MPI event in the trace that is to be processed from the queues. A single rank MPI event is an MPI event that only involves one thread.</td>
</tr>
<tr>
<td>FuncEventInfo * <em>event</em> ) = 0;`</td>
<td></td>
</tr>
<tr>
<td>`virtual void ProcessCollOp (</td>
<td>Your simulator needs to implement this API function. This function processes a collective operation MPI event in trace file. This event object is a collective operation MPI event in the trace that is ready to be processed from the queues. This one object represents all threads involved in collective operation. The simulator does not get separate events for each thread involved in the collective operation.</td>
</tr>
<tr>
<td>CollOpInfo * <em>collop</em> ) = 0;)`</td>
<td></td>
</tr>
<tr>
<td>`virtual void ProcessP2P ( P2PInfo *</td>
<td>Your simulator needs to implement this API function. This function processes a point-to-point MPI event in trace file. This event object is a P2P MPI event in the trace that is to be processed from the queues. This function is called for sendrecv, synchronous, or normal send to receive MPI operations.</td>
</tr>
<tr>
<td><em>message</em> ) = 0;)`</td>
<td></td>
</tr>
<tr>
<td>`virtual void ProcessNonBlockingSend (</td>
<td>Your simulator needs to implement this API function. This function processes a non-blocking send MPI event in a trace file. This function gets an event object that represents the send operation and the corresponding P2P message that goes with that send. This function is only called for non-synchronous sends, or sends that do not rely on the receive thread. The start time of the receive thread may not be known.</td>
</tr>
<tr>
<td>FuncEventInfo * <em>event</em>, P2PInfo * <em>message</em> ) =0;`</td>
<td></td>
</tr>
</tbody>
</table>
virtual void ProcessP2PWaitAllSome ( Vector<P2PInfo*, ObjectPointerAllocator<P2PInfo*> > * _messages_ ) = 0;

Your simulator needs to implement this API function. This function processes a waitall and waitsome MPI event in trace file. This function gets a vector of all P2P messages involved with the waitall/waitsome. There is no individual function events corresponding to the send or receive thread involved in this P2P operation.

### 10.2.3.6. Private Members

<table>
<thead>
<tr>
<th>Member</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector&lt; char* &gt; _commandLineParams;</td>
<td>This vector represents the command-line parameters passed to the simulator by CPF. It is a vector of char pointers of parameters that do not match any keywords for CPF. These parameters may contain invalid values, and need to be checked by simulator.</td>
</tr>
<tr>
<td>char * _stfPath;</td>
<td>Pathway to the directory of the original STF file loaded by CPF at runtime.</td>
</tr>
<tr>
<td>U8 _stfStart;</td>
<td>Start time of original STF file.</td>
</tr>
<tr>
<td>U8 _stfEnd;</td>
<td>End time of original STF file</td>
</tr>
<tr>
<td>U4 _numThreads;</td>
<td>Number of threads in trace</td>
</tr>
<tr>
<td>F8 _clockResolution;</td>
<td>Resolution length of clock tick in seconds.</td>
</tr>
<tr>
<td>Vector&lt; U4 &gt; _singleRankFuncs;</td>
<td>Vector of single rank function id's. This vector is initialized in an init() call with the id values of every rank function in the trace file. Ranks that are not in the trace file are initialized with the value -1.</td>
</tr>
<tr>
<td>Vector&lt; U4 &gt; _collOpFuncs;</td>
<td>Vector of collective operation id's. This vector is initialized in an init() call with the id values of every collective function in the trace file. Given a function id number, the corresponding collective operation id is returned.</td>
</tr>
<tr>
<td>Vector&lt; U4 &gt; _P2PFuncs;</td>
<td>Vector of P2P identifiers. This vector is initialized in an init() call with the id values of every P2P function in the trace file. This vector returns the related the corresponding collective operation id for every function id number.</td>
</tr>
<tr>
<td>HashMap&lt; U4, const char* &gt; _communicatorsMapped;</td>
<td>HashMap of all communicator names. Lookup is by id of communicator. Data value is strings.</td>
</tr>
</tbody>
</table>
10.2.3.7. Private Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bool isSingleRankFunc( U4 <em>id</em> )</td>
<td>Checks to see whether the function id is a single rank ID. Returns TRUE if <em>id</em> is a single rank function; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isCollectiveOperation( U4 <em>id</em> )</td>
<td>Checks to see whether the function id matches a collective operation ID. Returns TRUE if <em>id</em> is a collective operation; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isP2P( U4 <em>id</em> )</td>
<td>This function checks to see whether the function id is a P2P ID. Returns TRUE if <em>id</em> is a P2P function; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isAllToAll( U4 <em>id</em> )</td>
<td>Returns TRUE if it is an ALL-TO-ALL collective operation id number; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isOneToAll( U4 <em>id</em> )</td>
<td>Returns TRUE if it is a ONE-TO-ALL collective operation id number; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isAllToOne( U4 <em>id</em> )</td>
<td>Returns TRUE if it is an ALL-TO-ONE collective operation id number; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isAll( U4 <em>id</em> )</td>
<td>Returns TRUE if it is an ALL collective operation id number; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isSendP2P( U4 <em>id</em> )</td>
<td>Returns TRUE when the FuncEvent data is a P2P send operation; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isSyncSendP2P( U4 <em>id</em> )</td>
<td>Returns TRUE when the FuncEvent data is a P2P sync send operation; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isSendRecv( U4 <em>id</em> )</td>
<td>Returns TRUE when the FuncEvent data is a P2P send-receive operation; FALSE otherwise.</td>
</tr>
<tr>
<td>bool isRecv( U4 <em>id</em> )</td>
<td>Returns TRUE when the FuncEvent data is a P2P receive operation; FALSE otherwise.</td>
</tr>
</tbody>
</table>

10.2.3.8. Typedef Functions

These functions are essential to providing the plug-in module for CPF. They are already included in DevSim and if you modified DevSim, then you should have properly modified these functions to handle your simulator. The `create_t` function creates an instance of your simulator class and passes a reference back to CPF. When CPF has completed processing a trace, it destroys the instance of the simulator through the `destroy_t` function.

typedef CustomPluginFramework* create_t();
typedef void destroy_t(CustomPluginFramework*);
11. Intel® Trace Collector Configuration Assistant

This topic provides introduction and detailed information on the Intel® Trace Collector configuration assistant. See the following topics:

1. General Description
2. Detailed Description

11.1. General Description

Intel® Trace Collector configuration assistant provides a simple GUI interface to view and edit the Intel Trace Collector configuration files. It can also compute an average size of the trace file taking into account the modified parameters.

Intel® Trace Collector Configuration Assistant

To launch the Intel® Trace Collector configuration assistant for trace.stf, type the following in the command line:

```
itrconfig trace.stf [configfile.conf]
```

The optional parameter `configfile.conf` is a valid Intel Trace Collector configuration file. If it is not presented in the command line, then the default settings will be used.

The configuration file data created by the assistant can be imported into the Intel Trace Collector when running an application linked with it.

11.2. Detailed Description

The assistant GUI has three panels:
Intel® Trace Collector Configuration Assistant

1. The left one contains the variable groups
2. The middle one contains the variables for each group
3. The right one contains the documentation for each group of variables

The status bar has two sections:
1. The left one represents the estimated trace file size for the currently set values of all the variables
2. The right one shows the configuration file name currently open or default if the assistant is working with the default configuration

Here is the list of the supported Intel® Trace Collector variables:

ACTIVITY MEM-MINBLOCKS
AUTOFLUSH MEM-OVERWRITE
COUNTER OS-COUNTER-DELAY
CURRENT-DIR PCTRACE
DEMANGLE PCTRACE-CACHE
DETAILED-STATES PCTRACE-FAST
ENTER-USERCODE PROCESS
ENVIRONMENT PROGNAME
FLUSH-PID PROTOFILE-NAME
FLUSH-PREFIX STATE
INTERNAL-MPI STATISTICS
LOGFILE-FORMAT STF-CHUNKSIZE
LOGFILE-NAME STF-PROCS-PER-FILE
LOGFILE-PREFIX STF-USE-HW-STRUCTURE
LOGFILE-RANK SYMBOL
MEM-BLOCKSIZE SYNCED-CLUSTER
MEM-FLUSHBLOCKS SYNCED-HOST
MEM-INFO SYNC-MAX-DURATION
MEM-MAXBLOCKS VERBOSE
12. Concepts

This chapter contains explanations of the following abstract concepts of Intel® Trace Analyzer:

- **Level of detail**
- **Aggregation**
- **Advanced aggregation**
- **Tagging and filtering**

12.1. Level of Detail

Tracing all available events over time can generate billions of events even for a moderate program runtime of a few minutes and a handful of CPUs. The sheer amount of data is a challenge for any analysis tool that has to cope with this data. This is even worse as in most cases the analysis tool cannot make use of the same system resources as the parallel computer on which the trace was generated.

An aspect of this problem arises when generating graphical diagrams of the event data. Obviously, it is next to impossible to graphically display all the data. Firstly, it would take ages to do that. Secondly, it would depend on round-off errors in the scaling and on the order of the data traversal which events would actually make it to the screen without being erased by others. So it is clear that only representatives of the actual events are shown.

A valid choice would be to paint only every 100th or 1000th event and to hope that the resulting diagram gives a valid impression of the data. But this approach has its problems, because the pattern selects the representatives can interfere with the patterns in the underlying data.

Intel® Trace Analyzer uses a Level of Detail concept to solve this problem. The Event Timeline Chart (as the other timelines) calculates a hint for the analysis that describes a time span that can reasonably be painted and selected with the mouse. This hint is called Resolution. The resolution requested by the timeline takes into account the currently available screen space and the length of the current time interval. Hence a higher screen resolution or a wider timeline results in more data being displayed for the same time interval.

Intel® Trace Analyzer then tries to find a near match for the requested resolution. The exact resolution depends on internals, which will not be discussed here.

Intel Trace Analyzer divides the requested time interval into slots of length resolution. After that, representatives for the function events, the messages and the collectives in these slots are chosen in a deterministic way. If a function spans more than the given resolution it results in a larger slot.

The representatives for function events are chosen as follows: for each slot and each process (or thread group respectively) there is only a single function event representing the function where the thread or group spent most of its time.

The representatives for messages are chosen as follows: for each tuple (sender, receiver, sender slot, receiver slot) only one message is generated that carries averaged attributes. These attributes are averaged over all messages matching the tuple.

The representatives for collective operations are chosen as follows: for each tuple (communicator, first slot) one collective operation is generated. So it can happen that an operation of type `MPI_Gather` is merged with an operation of type `MPI_Bcast` resulting in a merged operation with no particular type at all (mixed).
To prevent misconceptions, emphasis is given to the fact that the merging of events only applies to the timelines and not to the profiles. The profiles always show sums, minima, maxima or averages over the complete set of events. The calculation of these results does obviously not depend on the screen resolution.

### 12.2. Aggregation

Aggregation reduces the amount of data by aggregating events into thread groups and into function groups.

The following aggregation types are discussed in this topic:

- **Thread aggregation**
- **Function aggregation**

#### 12.2.1. Function Aggregation

Aggregation into function groups enables you to decide on what level of detail to look at the threads or thread groups' activity. In many cases it might be enough to see that a code spends some percent of its time in MPI without knowing in which particular function. In some cases optimizing the serial parts of the program might seem more rewarding than optimizing the communication structure.

However, if the fraction spent in MPI exceeds the expectation, then it is interesting to know in which particular MPI call the time was spent. Function grouping allows exactly this shift in perspective by ungrouping the function group MPI.

While the argumentation given in [Thread Aggregation](#) for having nested thread groups may not be that compelling, the reason for having nested function groups comes quite clear as soon as there occur nested modules, classes and/or name spaces. It gets clearer if the binary instrumentation feature of Intel® Trace Collector is used as the result is thousands of functions instrumented.

Provided that there are adequate function groups, it is also much easier to categorize code by library or by author. In this way, it is possible to concentrate precisely on the code that is considered tunable while code that is controlled by third parties is aggregated into coarse categories.

---

**NOTE**

Selecting a function group generally results in displaying the information for the group's children. That is the reason why single functions cannot be selected for aggregation.

In the case of timelines, certain events may not be visible at all times. It does not necessarily mean that they are not there. It can happen because the finer your grouping is, the less time is spent in each individual function/group. On aggregating over processes in a time interval where some processes are idle, nothing is displayed because of the idle state of the processes. Zooming in helps to see these events better. For a better overview, check one of the corresponding profiles. If the timeline and the profile seem to contradict, then the information from the profile is more precise.
12.2.2. Thread Aggregation

A striking example for the benefit of thread groups is a parallel code that runs on a cluster of SMP systems. In fact this scenario was the inspiration to introduce this concept. To analyze the behavior of such an application, the data transfer rate is verified to check if the reached rate is plausible with respect to the data rates that are expected (maybe a fraction of the data rates advertised). Of course the effective and expected data transfer rates differ for messages that travel inside an SMP node (intra-node) and between two SMP nodes (inter-node).

In the Intel® Trace Analyzer selecting Aggregation into the predefined All_Nodes process group is enough to make the distinction between intra-node and inter-node messages very easy: in the Message Profile the values for the intra-node messages appear on the diagonal of the matrix.

**NOTE**

Selecting a process group generally results in displaying the information for the group children (with the notable exception of the function profile). That is the reason why you cannot select single, unthreaded processes or single threads for aggregation.

The hierarchy is quite complicated: threads living on the same core (due to Hyper Threading), threads living on different cores in the same CPU, threads living on the same FSB in different CPUs, threads living in the same SMP box on different FSBs, threads living in different boxes connected by a faster interconnect, threads living in SMP boxes connected by a not so fast interconnect and so on. But such hierarchies allow for deeply nested thread groups.

If you select the thread group representing a single node to concentrate on intra-node effects, then the analysis becomes slower than using the thread group All_Processes alone. Why does it happen? First of all, Intel Trace Analyzer does not have to do any aggregation for the All_Processes thread group because it is flat (assuming no threads are used). The second is, despite the fact that only a single SMP node is chosen, all other threads go through the analysis and are thrown into the artificially created thread group Other. Click on Advanced > Show Process Group 'Other' to make this group visible. To speed things up, choose a filter that only lets the threads of the selected SMP node pass.

**NOTE**

Filtering and Aggregation are orthogonal mechanisms in the Intel Trace Analyzer.

12.3. Advanced Aggregation

The advanced aggregation function is based on process aggregation and allows lightening the Event Timeline chart for big traces above 1K processes. By default, the function is automatically enabled for traces with above 768 processes, but you can change this threshold using the Edit Configuration Dialog Box.
When you enable this function, all processes are divided into a set of serial groups. Each group has a certain count of processes (except for the last one) and corresponds to a single vertical frame on the Event Timeline chart. The default frame size is 512 displayed processes. You can change this value in the Edit Configuration Dialog Box. These processes groups are called

**All_Processes_(Visible_N1-N2)**

1. \( N1 \) is the start of the displayed range of processes in the group
2. \( N2 \) is the end of the range

For example, the first advanced aggregation group is called **All_Processes** by default (**Visible_0-4**).

The advanced aggregation groups contain hidden (non-visible separately) processes as two additionally aggregated groups:

1. aggregated_above
2. aggregated_below

They are displayed on the top and on the bottom of a frame respectively.
A vertical scrollbar of **Event Timeline** chart enables you to scroll among all advanced aggregation frames.
NOTE

When a jump to the next frame happens, Intel Trace Analyzer loads the corresponding process group. The chart may be not available in a short period.

See Also
Process Aggregation
Event Timeline
Edit Configuration Dialog Box

12.4. Tagging and Filtering

Both concepts use the same grammar to describe their expressions. See Filtering Dialog Box and Tagging Dialog Box for usage hints.

Conceptually there is a different filter for each class of events: function events, messages and collective operations. During analysis the event is recognized when the expression is matched.

The behavior of a filter is determined at the time of its creation. A filter continues filtering until it is changed, even when the thread groups or function groups that it references are changed. Events are treated as belonging to these groups based on the state of the groups at the time the filter was first created.

See Also
Filtering Dialog Box
Tagging Dialog Box

12.4.1. Tagging

If several events are merged as described in the Level of Detail section, then the merged event is tagged if at least one of the singular events is tagged.

Therefore you can use tagging in particular together with the Qualitative Timeline Chart (see Qualitative Timeline) to find events matching a certain criterion. If a single event out of billions matches the criteria of the tag filter, you see a highlighted peak in the Qualitative Timeline Chart that indicates where to zoom into the trace.

See Also
Level of Detail
Qualitative Timeline

12.4.2. Filtering

If an event is suppressed by filtering then the effect is as if it were never written to the trace file. This is relatively easy to understand for messages and collective operations.

However, if a filter is designed in the way that it suppresses MPI and enables all other functions to pass, then after filtering the Event Timeline looks as if MPI was not called. It looks as if the thread was in the calling function instead of being in MPI. The same is true for the call tree and the call graph of the Function Profile.
What happens if there are functions A, B and C in the trace where A calls B and B calls C and you filter B out? It appears as if A had called C directly. This is quite different from choosing a function aggregation that does not cover B, because that shows the function group other (other fgroup) wherever B was shown before. Filtering and Aggregation are independent.