# Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.9</td>
<td>2012-09-07</td>
<td>Documentation update.</td>
</tr>
<tr>
<td>2.2.7</td>
<td>2012-10-15</td>
<td>Documentation update.</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Introduction .......................................................................................................................... 1  
   1.1 What is Synchro?.............................................................................................................. 1  
   1.2 What Synchro Does with Your Data? ............................................................................ 1  
   1.3 Why Use Synchro? ........................................................................................................ 2  
   1.4 Architecture Overview ................................................................................................. 3  
   1.5 Synchro Versions ........................................................................................................ 3  
2 SynchroWeb Interface ......................................................................................................... 4  
   2.1 Introduction ................................................................................................................ 4  
      2.1.1 Views ..................................................................................................................... 4  
      2.1.2 Selecting Metrics ................................................................................................. 4  
      2.1.3 Selecting Time ...................................................................................................... 5  
      2.1.4 Selecting Statistics ............................................................................................... 6  
   2.2 Basic Controls ............................................................................................................... 8  
   2.3 Window Controls .......................................................................................................... 8  
   2.4 Data Controls ................................................................................................................ 9  
   2.5 View Controls ............................................................................................................. 10  
   2.6 Formats Description .................................................................................................... 11  
      2.6.1 PeriodStatistics Format ...................................................................................... 11  
      2.6.2 ByTimeStatistics Format .................................................................................. 16  
      2.6.3 Miscellaneous Formats ...................................................................................... 19  
3 Synchro Administration ..................................................................................................... 21  
   3.1 SynchroServer Installation .......................................................................................... 21  
      3.1.1 Pre-Requisite ....................................................................................................... 21  
   3.2 Installation Instructions ............................................................................................... 21  
   3.3 SynchroServer Start/Stop Procedures ......................................................................... 22  
   3.4 SynchroClient Solaris Installation ............................................................................ 22  
   3.5 SynchroClient Linux Installation ............................................................................... 23  
   3.6 SynchroClient Windows Installation ........................................................................... 25  
4 SynchroServer Configuration ............................................................................................. 26  
   4.1 Basic Admin Controls ................................................................................................. 26  
   4.2 Action Menu ............................................................................................................... 26  
   4.3 SynchroServer Configuration .................................................................................... 27  
      4.3.1 Server - Configuration ....................................................................................... 27  
      4.3.2 Server - Retention .............................................................................................. 27  
      4.3.3 Server - Compression ......................................................................................... 28  
      4.3.4 Server - Aggregator ............................................................................................ 28  
      4.3.5 Server - Logging ................................................................................................. 28  
      4.3.6 Logs and Alarm Parameters ............................................................................... 28  
      4.3.7 Server - Internal Parameters ............................................................................. 29  
   4.4 Alarm Configuration .................................................................................................. 29  
   4.5 MAVG Configuration ................................................................................................. 30  
   4.6 Color Configuration ................................................................................................... 31  
   4.7 Users Configuration .................................................................................................. 31  
   4.8 Periods Configuration ............................................................................................... 31  
   4.9 Aggregator Configuration ......................................................................................... 32  
   4.10 Compression Configuration ...................................................................................... 32  
   4.11 Retention Configuration .......................................................................................... 33  
   4.12 Client Configuration ................................................................................................ 33  
   4.13 Dropfrom Configuration ........................................................................................... 33
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.14 Memory Trigger Exception Configuration</td>
<td>34</td>
</tr>
<tr>
<td>4.15 Time Series Mapping Configuration</td>
<td>34</td>
</tr>
<tr>
<td>4.16 Value Mapping Configuration</td>
<td>34</td>
</tr>
<tr>
<td>4.17 Presets Configuration</td>
<td>34</td>
</tr>
<tr>
<td>Appendix A. SynchroClients</td>
<td>36</td>
</tr>
<tr>
<td>A.1 SynchroLinux</td>
<td>36</td>
</tr>
<tr>
<td>A.2 SynchroWindows</td>
<td>36</td>
</tr>
<tr>
<td>Appendix B. Application Integration</td>
<td>37</td>
</tr>
<tr>
<td>B.1 Trivial TCP Text Protocol</td>
<td>37</td>
</tr>
<tr>
<td>B.2 SynchroTail</td>
<td>39</td>
</tr>
<tr>
<td>B.3 Integration Examples</td>
<td>40</td>
</tr>
<tr>
<td>Appendix C. SynchroServer Performance Tuning</td>
<td>47</td>
</tr>
<tr>
<td>C.1 Increase Disk Performance</td>
<td>47</td>
</tr>
<tr>
<td>C.2 Increase CPU Usage</td>
<td>47</td>
</tr>
<tr>
<td>C.3 Increase Memory Available</td>
<td>48</td>
</tr>
</tbody>
</table>
1 Introduction

This document provides the information required to use the Synchro software. Discrepancies in this documentation or bugs in the Synchro software should be reported to support@affatechnology.com (more information available at www.synchro-tech.com).

Thank you for using Synchro.

1.1 What is Synchro?

Synchro is a real time enterprise infrastructure-monitoring tool that acquires and manages data and events from multiple sources throughout the enterprise. Synchro collects, aggregates, historizes, alerts, visualises, and reports on your data.

Think of Synchro as a spreadsheet, only field values in Synchro can change every second, or if necessary, on a more frequent basis. Like a spreadsheet, Synchro is not aware of the meaning of the data it processes.

A field in Synchro, and the associated data that changes every second is called a Time Series. A typical Synchro infrastructure processes hundreds of thousands of these Time Series updates every second.

1.2 What Synchro Does with Your Data?

For each Time Series, Synchro provides two (2) types of statistics.

- ByTimeStatistics
- PeriodStatistics

For each second of the day, within each Time Series, Synchro provides the latest value (LAST), Min, Max, Average, Count, and Sum, of the data received for that second. These second level statistics are called ByTimeStatistics.

Additionally, in Synchro it is possible to define Periods. A Period describes a continuous interval for a time of the day. For each Period, and for each Time Series, Synchro provides the latest value (LAST), Min, Max, Average, Count, Sum, Time (time of the last update), and Dtime (time since the last update). These Period based statistics are called PeriodStatistics. By default, Synchro keeps hourly, morning, evening, work hours, and daily PeriodStatistics. Users can define as many (possibly overlapping), Periods as required.

Synchro can also “artificially” create Time Series based on your data. It can group them together, aggregate days of data together, create new Time Series using moving aggregation functions (like moving averages), and create alerts that they themselves, become Time Series.

Synchro uses four (4) identifiers to uniquely identify a Time Series. By convention, the identifiers, Types and Metrics define what is being measured, and the identifiers, Components and Targets identify what it is measured on.

One of the most useful and distinctive feature of Synchro is the possibility of using values of PeriodStatistics to select a Time Series (in addition to the name of a Time Series). This makes it easy and efficient to display a real-time list (updated every second), of the top “N” clients/products/systems, based on any Time Series.
Synchro also offers the possibility of comparing missing/added items between dates and periods of the day, making it a simple task to compare the configuration of systems to see what was changed.

1.3 Why Use Synchro?

Synchro is an outstanding monitoring and capacity-planning tool that has been successfully used in diverse contexts such as industrial infrastructure, technology infrastructure, and application monitoring.

Synchro is...

- **Efficient**: Each Synchro infrastructure can process a sustained load of more than 200,000 data updates per second on commodity server hardware.
- **Granular**: By default, all data in Synchro is kept at the second level precision. Data can be historized and efficiently at the second level precision for years if required.
- **Insightful**: By pro-actively re-calculating masses of statistics every second, and allowing for aggregation and smoothing of data, Synchro provides real-time answers to questions such as:
  - Did something change?
  - What was changed?
  - What was added?
  - What is missing?
  - What are the impacts of the change?
1.4 Architecture Overview

The SynchroServer, a multi-threaded JAVA application, is at the heart of the Synchro infrastructure. It receives data from probes, applications, and adapters to aggregate, historize, report, and create alarms as needed. The data is sent to the SynchroServer via a trivial text TCP protocol. The SynchroWEB and SynchroAdmin tools are used within a WEB browser to interact the SynchroServer embedded WEB server via the HTTP protocol.

![Architecture Overview Diagram]

1.5 Synchro Versions

There are two distinct versions of Synchro. There is the full Commercial version described in these pages, and the Demonstration version which is valid for three months. The following table outlines the differences between the versions.

For more information on how to acquire the Commercial version of Synchro, please contact sales@affatechnology.com.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Commercial Version</th>
<th>Demonstration Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>Historical Data</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>Alarms</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>Reports</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>All Others</td>
<td>Included</td>
<td>Included</td>
</tr>
</tbody>
</table>
2 SynchroWeb Interface

This section presents the interface used in SynchroWEB.

2.1 Introduction

The SynchroWEB interface is the main tool that users will utilize to interact with the SynchroServer. Users can connect and use the SynchroWeb interface from the following URL by using a web browser such as; Safari, Chrome, or Firefox:

http://ServerName:8080/Synchro/Synchro.html

2.1.1 Views

The SynchroWEB interface is built around the Views concept. Users can create Views that are composed of any number of Data Windows. Data Windows present a set of data in a diverse format (e.g., graphs, text, tables…). Views can be protected (read only) and, depending on the SynchroServer mode, can also be hidden to other users.

2.1.2 Selecting Metrics

To select which Time Series are to be reported on, Synchro uses PeriodStatistics.

The Search tool (查找), available in the Data Window/Data Controls, is used to search for Time Series. Using this tool, users can select dates, periods, and a set of Time Series to be displayed. The identifiers Component/Target and Type/Metrics are used to select Time Series based on their name. Using the sliders; Click, Ctrl-Click, and Alt-Click, users can select the Time Series to be displayed. Additionally, users can also enter a string in the overhead fields to match a set of Time Series. Strings are composed of regular expressions. Using this functionality, users can select many Time Series by entering only one string.

For example: The string “System[12345]|System[1.*1]” would match System 1 through 5 and System10A1.

The table below describes the basic functionalities of regular expressions.

<table>
<thead>
<tr>
<th>Character</th>
<th>Signification</th>
</tr>
</thead>
<tbody>
<tr>
<td>. (dot)</td>
<td>Matches any single character except line break characters \r and \n.</td>
</tr>
<tr>
<td>* (star)</td>
<td>Repeats the previous item zero or more times.</td>
</tr>
<tr>
<td>+ (plus)</td>
<td>Repeats the previous item once or more</td>
</tr>
<tr>
<td>\Q...\E</td>
<td>Matches the characters between \Q and \E literally</td>
</tr>
<tr>
<td>^ (caret)</td>
<td>Matches the start of the string the regex pattern is applied to.</td>
</tr>
<tr>
<td>$ (dollar)</td>
<td>Matches the end of the string the regex pattern is applied to.</td>
</tr>
<tr>
<td>[123]</td>
<td>Defines inside a character class.(Matches Character 1, 2 or 3).</td>
</tr>
<tr>
<td></td>
<td>Causes the regex engine to match either the part on the left side, or the part on the right side.</td>
</tr>
</tbody>
</table>

**Note:** More information on “regular expressions can” be found by searching the Internet.
Using the above selection scheme, it is possible to simultaneously select thousands of Time Series.

The following parameters are also used to refine which Time Series, and on what, will be reported.

The Function menu (的功能) used with the Sort Criteria (排序条件), and the Options (参数) /"Data and Range”/"Maximum Number of Time Series” are used to ultimately select which Time Series are to be reported on.

### 2.1.3 Selecting Time

The Search tool (搜索) enables the user to select which date(s) and period(s) are to be used for Time Based and Time Series selection.

The Time Sample (范围) is used to select the size of the Sliding Window that ranges from the current server time, to a specified duration in the past (in seconds). This functionality is also available in Options (参数)/"Data and Range”/"Time Sample”.

Users can also select a specific interval of time using Options (参数)/"Data and Range”/"ZoomFrom”, and “Data and Range”/"ZoomTo” (Format “HHMMSS”).

Dates can be selected using the format YYYYMMDD in the Search tool. Dates and Periods can also be relatively defined using the current SynchroServer Date/Period. Below are a few examples:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Today</td>
<td>D-00</td>
</tr>
<tr>
<td>Yesterday</td>
<td>D-01</td>
</tr>
<tr>
<td>117 Days ago</td>
<td>D-117</td>
</tr>
<tr>
<td>Last Business Day</td>
<td>W-01</td>
</tr>
<tr>
<td>Today and Yesterday</td>
<td>D-00</td>
</tr>
<tr>
<td>This Week Mon-Sun</td>
<td>WK-00</td>
</tr>
<tr>
<td>Last Week Mon-Sun</td>
<td>WW-00</td>
</tr>
<tr>
<td>Last Week Mon-Fri</td>
<td>WW-01</td>
</tr>
<tr>
<td>This Month Mon-Sun</td>
<td>MO-00</td>
</tr>
<tr>
<td>This Month Mon-Fri</td>
<td>MO-01</td>
</tr>
<tr>
<td>Last Month Mon-Fri</td>
<td>MN-01</td>
</tr>
<tr>
<td>ThisHour</td>
<td>PH-00</td>
</tr>
<tr>
<td>LastHour</td>
<td>PH-01</td>
</tr>
</tbody>
</table>
2.1.4 Selecting Statistics

For each Time Series Synchro stores two (2) types of statistics. There are specific Synchro Formats designed to report on the different statistics.

**Note:** When using statistics pay attention to the Function menu (↑↑↓↓) as this menu selects which aggregation function is to be used for reporting.

The **PeriodStatistics** are designed to provide an overview over a Period of time of the detailed second by second statistics. Users can define as many Periods as required. By default, Synchro provides several Periods as presented in the following table.

![PeriodStatistics Table](image)

For each second of the day, and for each Time Series, Synchro provides the Last (latest value), Min, Max, Average, Count, and Sum of the data received. These second level statistics are called **ByTimeStatistics**, and are designed to provide the level of detail required to drill down into events that require more attention.

The following table demonstrates that for each second, the SynchroServer received ~2246 updates (count) for this particular Time Series. Other statistics for the data received are also available.
The image contains two graphs showing data from a Synchro Server. The graphs are labeled as follows:

**Graph 1:**
- **SynchroServer:** Synchro-server1.synchro-tech.com
- **Component:** Synchro
- **Type:** SYNCHRO
- **Metric:** METRICRATE
- **Target:** Server
- **Date:** 2012-09-01
- **Period:** 00@24
- **Sort:** Metric

The table below the graph lists the time with corresponding values for Last, Max, Average, Min, Sum, and Count.

<table>
<thead>
<tr>
<th>Time</th>
<th>Last</th>
<th>Max</th>
<th>Average</th>
<th>Min</th>
<th>Sum</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:22:07</td>
<td>1.00</td>
<td>408.00</td>
<td>1.18</td>
<td>1.00</td>
<td>2863.00</td>
<td>2245</td>
</tr>
<tr>
<td>17:22:08</td>
<td>1.00</td>
<td>408.00</td>
<td>1.18</td>
<td>1.00</td>
<td>2863.00</td>
<td>2245</td>
</tr>
<tr>
<td>17:22:09</td>
<td>1.00</td>
<td>382.00</td>
<td>1.16</td>
<td>1.00</td>
<td>2827.00</td>
<td>2245</td>
</tr>
<tr>
<td>17:22:10</td>
<td>1.00</td>
<td>382.00</td>
<td>1.16</td>
<td>1.00</td>
<td>2827.00</td>
<td>2245</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Graph 2:**
- **SynchroServer:** Synchro-server1.synchro-tech.com
- **Component:** Synchro
- **Type:** SYNCHRO
- **Metric:** METRICRATE
- **Target:** Server
- **Date:** 2012-09-01
- **Period:** 00@24
- **Sort:** Metric
- **Func:** Metric

The graph shows a time series plot with the y-axis ranging from 0 to 3000. The x-axis represents time from 17:22 to 17:23.

- [Last] (left)
- [Average] (left)
- [Max] (left)
- [Min] (left)
- [Sum] (left)
- [Count] (left)
2.2 Basic Controls

The following basic controls are available in the upper portion of the SynchroWEB interface.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Home View Icon" /></td>
<td>The <strong>Home View</strong> icon is used to select the &quot;home&quot; view. The user uses Synchro to create the Home view (view name is home). If Home view is not defined, the “default” View is used.</td>
</tr>
<tr>
<td><img src="image" alt="Views Icon" /></td>
<td>The <strong>Views</strong> icon is used to access the <strong>View Selector</strong> interface.</td>
</tr>
<tr>
<td><img src="image" alt="Report Icon" /></td>
<td>The <strong>Report</strong> Icon is used to activate the <strong>Report Selector</strong> tool to view already created reports.</td>
</tr>
<tr>
<td><img src="image" alt="View Controls Icon" /></td>
<td><strong>View Controls</strong> provide basic view controls such as; Save, Save As, and Delete for the views. This is also the menu used to access the <strong>SynchroAdmin</strong> interface.</td>
</tr>
<tr>
<td><img src="image" alt="Lock/UnLock Icon" /></td>
<td>The <strong>Lock/UnLock</strong> icons show the status of the View. Only the owner of the View can lock or unlock the View.</td>
</tr>
</tbody>
</table>

In addition, Synchro displays the View name, and the name of the current user in the upper right portion of the screen.

2.3 Window Controls

Window Controls are available when the mouse pointer hovers on the upper right portion of any Data Window.

**Note:** The View needs to be Unlocked ( ![Unlock Icon](image) ) for the controls to appear.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Info Icon" /></td>
<td>Provides information about the status of the query sent to the server by the <strong>SynchroWEB</strong> interface.</td>
</tr>
<tr>
<td><img src="image" alt="Pause and Restart Icon" /></td>
<td><strong>Pause</strong> and <strong>Restart</strong> the update of the Data Window.</td>
</tr>
<tr>
<td><img src="image" alt="Duplicate Icon" /></td>
<td><strong>Duplicate</strong> (copy) the Data Window.</td>
</tr>
<tr>
<td><img src="image" alt="Export Icon" /></td>
<td><strong>Export</strong> the content of the Data Window. Export default format is MS Excel (xls). Format can be changed in &quot;View Controls ( <img src="image" alt="View Options Icon" /> )&quot;/&quot;View Options&quot;/&quot;ExportFormat&quot;</td>
</tr>
<tr>
<td><img src="image" alt="Close Icon" /></td>
<td><strong>Close Data Window</strong>. Note – users cannot delete all Data Windows – one Data Window must remain open.</td>
</tr>
</tbody>
</table>
## 2.4 Data Controls

Data Controls are available when the mouse pointer hovers on the upper left portion of any Data Window (below the **Title** region).

**Note:** The View needs to be Unlocked (ião) for the controls to appear.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td><strong>Manually Refresh</strong> the content of the Data Window when Paused.</td>
</tr>
<tr>
<td>🕵️</td>
<td><strong>Expand/Collapse</strong> the Data Window.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔍</td>
<td>The <strong>Search</strong> tool is used to select the <strong>Time Series</strong> to be reported on. In addition, sort criteria and the <strong>Options / “Data and Range”/”Maximum Number of Time Series”</strong> when used together, select which Time Series are to be reported on.</td>
</tr>
<tr>
<td>🔧</td>
<td>The <strong>Option</strong> tool provides access to the &quot;Window Control&quot;, &quot;Data and Range&quot;, and &quot;Format Specific” options.</td>
</tr>
<tr>
<td>⌚️</td>
<td>The <strong>Time Sample</strong> is used to select the size of the Sliding Window that ranges from the current server time to the specified duration in the past (in seconds). This functionality is also available with Options/&quot;Data and Range”/&quot;Time Sample”. The values available in this menu can be changed in the SynchroAdmin interface.</td>
</tr>
<tr>
<td>📆</td>
<td>Selects the <strong>Periods</strong> be used for data selection (PeriodStatistics) and for time selection. This functionality is also available in the Search tool. The values available in this menu can be changed in the SynchroAdmin interface.</td>
</tr>
<tr>
<td>📅</td>
<td>Selects the <strong>Dates</strong> to be used for data selection. This functionality is also available in the Search tool. The values available in this menu can be changed in the SynchroAdmin interface.</td>
</tr>
<tr>
<td>📊</td>
<td>Selects the <strong>Format</strong> of the data to be displayed. See Formats description below for details.</td>
</tr>
<tr>
<td>⤴️</td>
<td>The <strong>Sort Criteria</strong> is used to select which Time Series are to be displayed, and also establishes the order in which they are to be displayed. A reverse sort is also available in Options/&quot;Data and Range”/&quot;Reverse Sort&quot;.</td>
</tr>
<tr>
<td>📊</td>
<td>The <strong>Function</strong> tool is used to select which aggregation function to display. This functionality is also available in Options/&quot;Data and Range”/&quot;Function&quot;. Possible values are LAST, MIN, MAX, AVG, COUNT, and SUM</td>
</tr>
</tbody>
</table>
2.5 View Controls

View Controls are available in the upper right portion of the screen under the (⚙️) icon. The following menu appears when the mouse pointer cursor hovers over the icon.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Share…</td>
<td>Enables the user to create an URL where all Basic Controls are removed. This is useful to provide a simple “read only” interface for specific users.</td>
</tr>
<tr>
<td>Save…</td>
<td>Save the current View</td>
</tr>
<tr>
<td>Save As…</td>
<td>Save the current View using a new name.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete current View.</td>
</tr>
<tr>
<td>View Options</td>
<td>View Options present some controls over the View. More information is available below.</td>
</tr>
<tr>
<td>SynchroAdmin</td>
<td>This menu item launches the SynchroAdmin interface. This interface is the tool the Synchro Server Administrator uses to configure the Synchro Server.</td>
</tr>
<tr>
<td>Logout</td>
<td>Logout logs the user out. Login information is requested as needed when a user attempts to access a functionality that requires login information.</td>
</tr>
</tbody>
</table>

The View Options menu presents the following controls over the View:

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export Format</td>
<td>Selects which report format is to be used to export the data in &quot;Window Controls&quot;. The “Query URL Only” export format is used to export a query that can be used in another application such as; a spreadsheet, a browser, another SynchroWEB interface, or wget.</td>
</tr>
<tr>
<td>View Background Color</td>
<td>Sets the View Background Color in a format acceptable in HTML: i.e.: red, green, blue, #RRGGBB, rgb(0,0,0), and rgba(0,0,0,0))</td>
</tr>
<tr>
<td>BackGround Image URL</td>
<td>Set the background image. Example: <a href="http://SynchroServerName/download/Diapositive1.png">http://SynchroServerName/download/Diapositive1.png</a></td>
</tr>
<tr>
<td>Maximum Items Retrieved in a Search</td>
<td>Sets the maximum number of Time Series that can be extracted in a single search (default 1000).</td>
</tr>
<tr>
<td>Snap to Grid</td>
<td>Snap to Grid is useful when placing windows in a View.</td>
</tr>
<tr>
<td>Force Login</td>
<td>Forces the user to login to see a View.</td>
</tr>
<tr>
<td>Size Style</td>
<td>Size of a View can be either “Fixed Size”, where the view size is independent to the size of the browser screen, or “Resize to Fit” (default), where the View size and included items resize</td>
</tr>
</tbody>
</table>
### Function Icon | Description
--- | ---
 | automatically according the screen size.
View Width/Height | When "Fixed Size" Size Style is selected, these parameters determine the size of the View (unit pixels).

#### 2.6 Formats Description

There are three (3) types of formats in Synchro. Some formats are designed to report on **PeriodStatistics**, some on **ByTimeStatistics** and some others are designed to provide complementary information to Views.

**Note:** Unless stated, the following examples relate to a single Time Series.

The SYNCHRO/METRICRATE (TYPE/METRIC) below, describes the number of individual Time Series that updated the SynchroServer each second.

Notice that a particular date and period (H08 – from 08:00:00 to 08:59:59) was selected.

#### 2.6.1 PeriodStatistics Format

The following formats present PeriodStatistics in diverse form; some of the most important features of each format are presented.

**Statistics (HTML)**

Offers a view of the PeriodStatistics for a particular period.
Highlights:

- Function Selection – The Function menu (¶) selects which aggregation function to display.
- Dynamic color selection
- Compare Dates/Periods in “Option Window/Format Specific” (Date format YYYYMMDD) allows to compare statistics from two Dates/Periods. Using this feature, the system displays what Time Series are missing in RED and the ones that were added in YELLOW. All the other Time Series (normal) are displayed in the default color. The following example displays missing/added processes of a host on a particular period of two (2) days.

- Alarms can be displayed on the window in this format
• Displays descriptive statistics

**Metric Table (HTML)**

Displays a cross table of Metrics vs. the different dimensions of the Time Series. In this example, it displays the different Metrics available for a list of processes on a host.

**StatBarGraph**

Displays a dynamic histogram of selected statistics.
**TIP:** Use “WK-00” to select the current week of statistics, then “Sort/Metrics” to order them by name (Metric), and then “Options/Reverse Sort” to reorder them in increasing order.

**Dial (Graph)**

Displays a dynamic dial of selected statistics.
**TIP:** In Options, change the Min/Max of the dial and set the number of steps of the dial.

**Custom Text (Graph)**

This format enables the user to create a text-based window that can display text and embedded statistics.

This is an example of Custom Text Format.

*Synchro average value for the 20120827 was 1.17 for the day before it was 1.17.*

To do this, use the “Options/Format Specific/Custom Text” editor to write the text. Individual statistics can then be inserted into the text using the “Options/Format Specific/Show Result Table” in the format ${ColumnLine} as per the following example:
2.6.2 ByTimeStatistics Format

This section presents the format to report on, second by second.

Graph By Time (Graph)

This format (might be the most often used) displays the data against time for a series of Time Series.

Highlights:
• Can display several aggregate functions and Time Series simultaneously.
• Can overlap or in-line several days of the same Time Series.
• Supports seconds scale
• Supports different types of line/points
• Can group, second by second, any displayed Time Series (Data and Range/GroupByTime)

**Histogram Graph (Graph), Histogram (HTML), Cumulative Histogram (HTML)**

Displays a dynamic histogram of the selected data. It expresses the percentage of events for each class.

The following example represents the percentage of time that a particular process is using a certain percentage of CPU over several days.

![Histogram Graph Example](image)

**Highlights:**
• Use “Options/Format Specific/” Min/Max/Step to set the parameters of the histogram.

**Scatter Plot (Graph)**

Displays the value of one Time Series against the values of another. In this example, it displays the percentage of CPU used by a process, and CPU usage on the host.
Highlights:

• Supports different types of line/points

Data By Time (HTML)
Displays the data against time for Time Series. In this example, it displays the percentage of CPU used by a process, and CPU usage on the host.

Highlights:

• Displays descriptive statistics.
• Reverse time display
• Dynamic color selection
• Display Statistics
2.6.3 Miscellaneous Formats

These formats allow adding external content to the Views.

**Custom Image (Graph)**

Enables the user to add an external image from any WEB server by entering the URL of the image. This format is useful for adding graphs and logos to your Views. Use the “Options/Window Control/Update” to set the refresh rate in seconds to the desired frequency.

Highlights:

- Image can be stretched, cropped, or set to actual size.

**Custom URL (HTML)**

Enables the user to add the content of other web sites into a Synchro View by entering its URL in “Options/Format Specific/Custom URL” as follows. This format is useful when adding content from other WEB sites, and other SynchroServer to your Views. Use the “Options/Window Control/Update” to set the refresh rate in seconds to the desired frequency.
3 Synchro Administration

3.1 SynchroServer Installation

This section presents instructions to conduct the installation of the SynchroServer and SynchroClient software.

3.1.1 Pre-Requisite

SYSTEM: SynchroServer needs to be installed on a UNIX server (i.e., Linux SUSE, RedHat, Centos, Ubuntu..., or Solaris). It needs at least 1 core, 2GB of RAM, and enough disks to store data as per the retention policy.

A conformable configuration for SynchroServer would include 8 cores, 12 GB of RAM and SSD drives.

Note: Slower disks can be used to store historized data. Please refer to the “Synchro Performance Tuning” section for more information.

User: To install SynchroServer, the SynchroServer Administrator needs to decide on a user that will own the software installation, data and processes. Typically a dedicated user gets created for this purpose (for example: synchro, synchrodev synchroprod...). The user must have access to enough data to keep historical data according to the retention policy. Also, the user needs access to Java 6+ to run Synchro. Make sure Java is in the user's path.

```
synchro@SynchroHost:~$ java -version
java version "1.6.0_16"
Java(TM) SE Runtime Environment (build 1.6.0_16-b01)
Java HotSpot(TM) Server VM (build 14.2-b01, mixed mode)
synchro@SynchroHost:~$ which java
```

3.2 Installation Instructions

1. Make the package Executable

```
synchro@host:~$ chmod +x ./SynchroServer-2.0.4.bin
```

2. Run the Setup Program

```
synchro@host:~$ ./SynchroServer-2.0.4.bin
LICENSE AGREEMENT
IMPORTANT – PLEASE READ THIS FIRST. THIS IS A LICENSE AGREEMENT.

AFFA TECHNOLOGY INC. ("AFFA") IS WILLING TO LICENSE THE
SYCHRO SOFTWARE (THE “SOFTWARE”) YOU (HEREAFTER REFERRED
TO AS “YOU”) ONLY UPON THE TERMS AND CONDITIONS CONTAINED
IN THIS LICENSE AGREEMENT (“AGREEMENT”).

---

Note: The license agreement will only appear on the demonstration version.

---

Do you accept the license agreement? [yes/no]
yes
Licence accepted, installing SynchroServer...
Backing up SynchroServer/www/Synchro to backup/20121011.135513
Backing up SynchroServer/www/SynchroAdmin to backup/20121011.135513
Backing up SynchroServer/www/SynchroDashBoard to backup/20121011.135513
Backing up SynchroServer/www/SynchroPlayer to backup/20121011.135513
Extracting and running setup.sh...
Replacing /SYNCHRO-INSTALL-DIR/ by /home/synchro/ in config files.
Modifying file: /home/synchro/SynchroServer/bin/start-server.sh
Modifying file: /home/synchro/SynchroServer/bin/stop-server.sh
Modifying file: /home/synchro/SynchroServer/bin/synchro-server.sh
Modifying file: /home/synchro/SynchroServer/bin/synchro-client.sh
Modifying file: /home/synchro/SynchroServer/bin/start-client.sh
Modifying file: /home/synchro/SynchroServer/bin/stop-client.sh
Modifying file: /home/synchro/SynchroServer/bin/synchrojmx
Modifying file: /home/synchro/SynchroServer/conf/server.ini
Done.

This will be used to configure remote SynchroClients
What is the name of this server [frean01]? 

Preparing SynchroServer/dist/server.conf...
Verifying java installation
All done, installation complete.

SynchroServer is now installed. You can now connect and use the SynchroWeb interface at the following URL (from a web browser e.g, IE or Firefox):

http://ServerName:8080/Synchro/Synchro.html

3.3 SynchroServer Start/Stop Procedures

To START the SynchroServer:

```bash
synchro@SynchroHost:~$ cd
synchro@SynchroHost:~$ : SynchroServer/bin/start-server.sh
SynchroClient stopped.
SynchroServer started.
synchro@SynchroHost:~$
```

To STOP the SynchroServer:

```bash
synchro@SynchroHost:~$ cd
synchro@SynchroHost:~$ : SynchroServer/bin/stop-server.sh
SynchroClient stopped.
SynchroServer stopped.
synchro@SynchroHost:~$
```

3.4 SynchroClient Solaris Installation

Download Synchro – you can download and install as many as needed. They will connect automatically to the SynchroServer they were downloaded from.

The user that will run the SynchroClients needs access to Java 6+ to run Synchro. Make sure Java is in the user's path. This is not a pre-requisite for the Windows Synchro Client.

```bash
synchro@SynchroHost:~$ java -version
java version "1.6.0_16"
Java(TM) SE Runtime Environment (build 1.6.0_16-b01)
Java HotSpot(TM) Server VM (build 14.2-b01, mixed mode)
synchro@SynchroHost:~$ which java
```

Download the Solaris SynchroClient

```bash
wget http://ServerName:8080/download/SynchroSolaris.tar.gz
```

```
prompt $ wget http://ServerName:8080/download/SynchroSolaris.tar.gz
--2009-07-02 15:37:49-- http://ServerName:8080/download/SynchroSolaris.tar.gz
Resolving ServerName... xxx.xx.26.112
Connecting to ServerName [xxx.xx.26.112]:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 985896 (963K) [application/x-gzip]
Saving to: `SynchroSolaris.tar.gz'
```
100%[==================================] 985,896 --.-K/s in 0.006s

2009-07-02 13:37:49 (151 MB/s) - "SynchroSolaris.tar.gz" saved [985896/985896]

# Untar the Client
prompt $ tar -xvf SynchroSolaris.tar.gz
SynchroSolaris/
SynchroSolaris/updateclient.sh
SynchroSolaris/bin/
SynchroSolaris/bin/SynchroClient.jar
SynchroSolaris/bin/logging.conf.xml
SynchroSolaris/bin/start-client.sh~
SynchroSolaris/bin/start-client.sh
SynchroSolaris/lib/
SynchroSolaris/lib/slf4j-api-1.4.3.jar
SynchroSolaris/lib/logback-core-0.9.8.jar
SynchroSolaris/lib/mina-core-1.1.7.jar
SynchroSolaris/lib/snmp4j.jar
SynchroSolaris/lib/logback-classic-0.9.8.jar
SynchroSolaris/README

# Start the Client
prompt $ SynchroSolaris/bin/start-client.sh
Starting: SolarisStatThreadNew

# See the interface in a web browser (IE, Chrome, Firefox...) using the URL below
http://ServerName:8080/Synchro/Synchro.html

Use the "Views" menu (upper right of the screen) to select what you want to see.

To START the SynchroClient:

```
synchro@SynchroHost:~$ cd
synchro@SynchroHost:~$ : SynchroLinux/bin/start-client.sh
SynchroClient stopped.
synchro@SynchroHost:~$
```

To STOP the SynchroClient:

```
synchro@SynchroHost:~$ cd
synchro@SynchroHost:~$ : SynchroLinux/bin/stop-client.sh
SynchroClient stopped.
synchro@SynchroHost:~$
```

### 3.5 SynchroClient Linux Installation

Download Synchro – you can download and install as many as needed. They will connect automatically to the SynchroServer they were downloaded from.

The user that will run the SynchroClients requires access to Java 6+ to run Synchro. Make sure Java is in the user’s path. This is not a pre-requisite for the Windows Synchro Client.

```
synchro@SynchroHost:~$ java -version
java version "1.6.0_16"
Java(TM) SE Runtime Environment (build 1.6.0_16-b01)
Java HotSpot(TM) Server VM (build 14.2-b01, mixed mode)
synchro@SynchroHost:~$ which java
```

Download the Linux SynchroClient

```
wget http://ServerName:8080/download/SynchroLinux.tar.gz
```

```
prompt $ wget http://ServerName:8080/download/SynchroLinux.tar.gz
Resolving ServerName... xxx.xx.26.112
Connecting to ServerName [xxx.xx.26.112]:8080... connected.
```
HTTP request sent, awaiting response... 200 OK
Length: 985896 (963K) [application/x-gzip]
Saving to: 'SynchroLinux.tar.gz'

100%[==========================================] 985896 985,896K/s in 0.006s
2009-07-02 13:37:49 (151 MB/s) - 'SynchroLinux.tar.gz' saved [985896/985896]

# Untar the Client
prompt $ tar -zvxf SynchroLinux.tar.gz
SynchroLinux/
SynchroLinux /updateclient.sh
SynchroLinux /bin/
SynchroLinux /bin/SynchroClient.jar
SynchroLinux /bin/logging.conf.xml
SynchroLinux /bin/start-client.sh~
SynchroLinux /bin/start-client.sh
SynchroLinux /lib/
SynchroLinux /lib/slf4j-api-1.4.3.jar
SynchroLinux /lib/logback-core-0.9.8.jar
SynchroLinux /lib/mina-core-1.1.7.jar
SynchroLinux /lib/snmp4j.jar
SynchroLinux /lib/logback-classic-0.9.8.jar
SynchroLinux /README

# Start the Client
prompt $ SynchroLinux /bin/start-client.sh
Starting:  LinuxStatThreadNew

# See the interface in a web browser (IE, Chrome, Firefox...) using the URL below
http://ServerName:8080/Synchro/Synchro.html

Use the "Views" menu (upper right of the screen) to select what you want to see.

To START the SynchroClient:

```
synchro@SynchroHost:~$ cd
synchro@SynchroHost:~$ : SynchroLinux/bin/start-client.sh
SynchroClient stopped.
SynchroServer started.
synchro@SynchroHost:~$
```

To STOP the SynchroLinux:

```
synchro@SynchroHost:~$ cd
synchro@SynchroHost:~$ : SynchroLinux/bin/stop-client.sh
SynchroClient stopped.
SynchroServer stopped.
synchro@SynchroHost:~$
```
3.6 SynchroClient Windows Installation

Download Synchro – you can download and install as many as needed. They will connect automatically to the SynchroServer they were downloaded from.

Monitoring a Windows Host (in a browser)

http://ServerName:8080/download/SynchroClient.exe
prompt & wget http://ServerName:8080/download/SynchroSolaris.tar.gz

Execute the installation package. The Synchro client will install itself as a WINDOWS service.

It is possible to deploy the SynchroWindows without having the installation script displaying the notification window. For this just a “–quiet” on the command line.
4 SynchroServer Configuration

This section describes the SynchroServer configuration. The SynchroAdmin interface is the tool the Synchro Server Administrator uses to configure the Synchro Server.

It can be accessed via the SynchroWeb interface under the “View Control”/ètres. Configuration parameters are saved in the <SynchroServer/conf> directory. These configurations are not affected by the installation or upgrade process.

4.1 Basic Admin Controls

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The Action menu provides functionality like Stop the Server, Open and Close the Statistics ports…</td>
</tr>
<tr>
<td></td>
<td>All SynchroServer configurations can be set in this menu. This includes the SynchroServer configuration, Users, Moving Averages, Alarms, Periods…</td>
</tr>
<tr>
<td></td>
<td>Let the user defined the content of the “Window Data Controls” presets</td>
</tr>
</tbody>
</table>

4.2 Action Menu

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop Server</td>
<td>Stops The SynchroServer</td>
</tr>
<tr>
<td>Garbage Collect</td>
<td>Manually run the Java JVM garbage collection process</td>
</tr>
<tr>
<td>Close Stat Port</td>
<td>Close the Statics acquiring port thus disconnection all probed and agents – only the query port will remain open (For SynchroWEB and SynchroAdmin interfaces).</td>
</tr>
<tr>
<td>Open Stat Port</td>
<td>Opens that statistics collection port permitting clients and probes to connect.</td>
</tr>
<tr>
<td>Load Layouts</td>
<td>Manually Load the Views Layouts from disk without rebooting the server.</td>
</tr>
<tr>
<td>Load Reports</td>
<td>Manually Load the Reports from disk without rebooting the server.</td>
</tr>
<tr>
<td>Load Presets</td>
<td>Manually Load the Data Controls Presets from disk without rebooting the server.</td>
</tr>
<tr>
<td>Run Retention</td>
<td>Manually run the retention process</td>
</tr>
<tr>
<td>Run Compression</td>
<td>Manually run the compression process</td>
</tr>
<tr>
<td>Run Reports</td>
<td>Manually run the report creation process.</td>
</tr>
</tbody>
</table>
4.3 SynchroServer Configuration

This section presents the most important configuration parameters of the SynchroServer.

4.3.1 Server - Configuration

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchro Title</td>
<td>Sets the SynchroServer name that appears in the title of Data Windows</td>
</tr>
<tr>
<td>Host Name</td>
<td>Is the host name or IP address of the host that runs the SynchroServer – this parameter is inserted in the SynchroClients packages so that they will connect to the SynchroServer without having to set the parameter each time.</td>
</tr>
<tr>
<td>Stat Port</td>
<td>TCP port that is used by clients and probes to connect to the SynchroServer and send Time Series statistics.</td>
</tr>
<tr>
<td>HTTP Port</td>
<td>Port used by SynchroWEB and SynchroAdmin to query the SynchroServer.</td>
</tr>
<tr>
<td>Lock View</td>
<td>Parameter that sets the SynchroServer in a mode where a login is required to access any View</td>
</tr>
<tr>
<td>Memory Allocation</td>
<td>The amount of memory available to the SynchroServer is fixed at start time. By default, the SynchroServer will not use more than 60% of available physical memory available on the host running the server. This can be changed using this parameter. (unit: Megabyte)</td>
</tr>
</tbody>
</table>

4.3.2 Server – Retention

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Execute Time</td>
<td>Sets the execute time of the retention utility on the SynchroServer Data</td>
</tr>
<tr>
<td>Log Execute Time</td>
<td>Sets the execute time of the retention utility on the SynchroServer Logs</td>
</tr>
<tr>
<td>Log Nb Days</td>
<td>Sets the number of days the SynchroServer will keep its internal logs</td>
</tr>
<tr>
<td>Report Execute Time</td>
<td>Sets the execute time of the report utility</td>
</tr>
<tr>
<td>Report Nb Days</td>
<td>Sets the number of days the SynchroServer will keep the reports it created</td>
</tr>
</tbody>
</table>
4.3.3 Server – Compression

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Time</td>
<td>Sets the execute time of the compression utility</td>
</tr>
</tbody>
</table>

4.3.4 Server – Aggregator

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregator Time</td>
<td>Sets the execute time of the Aggregator utility</td>
</tr>
</tbody>
</table>

4.3.5 Server – Logging

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Synchro Log</td>
<td>Determine if SynchroServer will create logs</td>
</tr>
<tr>
<td>Synchro Log Format</td>
<td>Sets Synchro Log formats String – See below for the parameters</td>
</tr>
<tr>
<td>Use Syslog</td>
<td>Determine if SynchroServer will use the Unix Syslog</td>
</tr>
<tr>
<td>Syslog Format</td>
<td>Sets the Syslog formats String – See below for the parameters</td>
</tr>
<tr>
<td>Syslog Facility</td>
<td>Sets the Syslog facility to use</td>
</tr>
<tr>
<td>Syslog Path</td>
<td>Path of the Syslog logd</td>
</tr>
<tr>
<td>Syslog Protocol</td>
<td>Sets the Syslog access mode</td>
</tr>
<tr>
<td>Syslog Host/Port</td>
<td>Sets the Host and the Port to access the Syslog</td>
</tr>
<tr>
<td>Use Email</td>
<td>Determine if SynchroServer will use emails to send alarms</td>
</tr>
<tr>
<td>Email Log Format</td>
<td>Sets the Email alarm formats String – See below for the parameters</td>
</tr>
<tr>
<td>Email From</td>
<td>Sets the address of the sender</td>
</tr>
<tr>
<td>Smart SMTP Server</td>
<td>Name or IP of the Email server</td>
</tr>
<tr>
<td>Smart SMTP Server Port</td>
<td>Port to use to connect to the Email server to send Alarms</td>
</tr>
</tbody>
</table>

4.3.6 Logs and Alarm Parameters

- %D: Date
- %A: alarm name
- %f: feed time HHMMSS
- %F: feed time HH:MM:SS
- %p: SynchroServer pid
%C: Component
%T: Type
%M: Metric
%t: target
%d: dt ime
%l: last
%m: max
%a: average
%n: min
%s: sum
%c: count
%O: old alarm value
%N: new alarm value
%L: alarm value in syslog severity level: INFO, WARN, CRIT, FAIL

4.3.7 Server - Internal Parameters

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge Engine Count</td>
<td>Number of internal engine that processes incoming Statistics from clients and probes</td>
</tr>
<tr>
<td>Result Engine Count</td>
<td>Number of internal engine that processes incoming HTTP queries from SynchroWEB interfaces</td>
</tr>
<tr>
<td>Freemem Trigger Level</td>
<td>Percentage of free memory threshold after which the SynchroServer switched into memory protect mode and disconnects all clients and probe leaving only the HTTP port open. Note that a disconnection exception list can be configured – See Memory Trigger Exception below</td>
</tr>
<tr>
<td>GCRuntime Trigger Level</td>
<td>Duration in ms of the Java Garbage Collect process after which the SynchroServer switched into memory protect mode and disconnects all clients and probe leaving only the HTTP port open. Note that a disconnection exception list can be configured – See Memory Trigger Exception below</td>
</tr>
<tr>
<td>Synchro Internal Metrics</td>
<td>Sets the level of detail of the SynchroServer logs.</td>
</tr>
</tbody>
</table>

4.4 Alarm Configuration

Synchro can generate alarms on PeriodStatistics of any Time Series based on pre-defined thresholds.

Alarm level range values are defined as; 0=Normal, 1=Warning, 2=Critical, 3=Fatal, 3+=Fatal. Alarms can be set on the following statistical aggregation functions of any Time
Series for a specific period: LAST, AVERAGE, LAST, MAX, MIN, DTIME (Delta in seconds since last update). Each Alarm in Synchro produces a Time Series named after the name of the alarm. This enables the user to track and report on the resulting Time Series as for any other Time Series.

ALARMS INTEGRATION

The alarm engine produces a log of the activity of the Alarm Time Series. Alarm logs can be found here: SynchroServer/log/AlarmLog-<Date>.log

It is also possible to use Syslog and Email for integration. Syslog and Email configuration can be configured in the Server Configuration tab of the SynchroAdmin. The string that defines the alarm can be changed, as needed – See Alarm string definition and parameters in the SynchroServer parameters.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the Alarm</td>
</tr>
<tr>
<td>Filter</td>
<td>Set the Time Series on which the alarm will be defined</td>
</tr>
<tr>
<td>Email</td>
<td>List of Email addresses to send the Alarm</td>
</tr>
<tr>
<td>Function</td>
<td>Aggregation function t set the Alarm -</td>
</tr>
<tr>
<td>OK/Warn/Critical/Fatal Trigger Level</td>
<td>Sets the minimum value after which an Alarm of the respective lever will be raised</td>
</tr>
<tr>
<td>Trigger Alarm On</td>
<td>Sets if an Alarm will be raised when the Alarm level augments, diminishes or both</td>
</tr>
<tr>
<td>Trigger Alarm if level equal or higher to</td>
<td>Sets the minimum level after which and Alarm is raised</td>
</tr>
</tbody>
</table>

4.5 MAVG Configuration

Any number of dynamic moving averages can be defined. The resulting Time Series name becomes the Metric name. It is possible to create alarms on the Moving average Time Series instead on the underlying data. This is useful to create less sensitive alarms.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the moving average</td>
</tr>
<tr>
<td>Duration</td>
<td>Length in seconds of the moving average</td>
</tr>
<tr>
<td>Filter</td>
<td>Set the Time Series on which the moving average will be defined</td>
</tr>
<tr>
<td>Algo</td>
<td>Selects a moving average algorithm</td>
</tr>
<tr>
<td></td>
<td>SumDivCount uses avg = sum of value divided by count of value received (doesn't count NA or missing values)</td>
</tr>
<tr>
<td></td>
<td>SumDivDuration uses avg = sum/duration so NA are simulated as 0</td>
</tr>
</tbody>
</table>
4.6 Color Configuration

Configuration parameters for:

- Background: Color of the Data Window Background
- Data: Color of Data displayed in Statistics and DataByTime.
- Line: Color of line to be displayed in GraphByTime
- Static: Color of Non dynamic color.

Colors supported also “Hide” and “Transparent”.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Defines a range of values that will use the color defined</td>
</tr>
<tr>
<td>Description</td>
<td>Name of the range – this name may be applied to the data in some cases</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>Minimum value that will use the color</td>
</tr>
<tr>
<td>Color</td>
<td>Color definition acceptable in HTML: ie: red, green, blue, #RRGGBB, rgb(0,0,0), rgba(0,0,0,0))</td>
</tr>
</tbody>
</table>

4.7 Users Configuration

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Name</td>
<td>Full Name of the user</td>
</tr>
<tr>
<td>User Name</td>
<td>User name that will be used for login</td>
</tr>
<tr>
<td>UID</td>
<td>User identification – generated by the SynchroServer</td>
</tr>
<tr>
<td>GID</td>
<td>Group ID – Users in group “1” gets their selected data window generated as reports.</td>
</tr>
<tr>
<td>Report Time</td>
<td>Time at which the reports are generated for this particular user</td>
</tr>
<tr>
<td>Read-Only Account</td>
<td>Sets the account as read only – so its can't create or modify Views</td>
</tr>
<tr>
<td>Read views of users</td>
<td>Sets that account so it can only see the Views created by a specific set of users</td>
</tr>
</tbody>
</table>

4.8 Periods Configuration

In Synchro it is possible to define Periods. A Period describes a continuous interval of time of the day. For each Period and for each Time Series, Synchro provides the Last (Last value), Min, Max, Average, Count, Sum, Time (Time of last update), DTimes (Time since last update). These Periods based statistics are called PeriodStatistics. By default, Synchro keeps hourly, morning, evening, work hours, and daily PeriodStatistics. Users can define as many, possibly overlapping, Periods as needed.
### 4.9 Aggregator Configuration

Synchro can group together several days of data for particular Time Series. At a time set in the SynchroServer configuration, the Aggregator reads historical data from Time Series to create a new one that represents the historical data.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the Period</td>
</tr>
<tr>
<td>Start Time</td>
<td>Start Time – Format: HHMMSS</td>
</tr>
<tr>
<td>End Time</td>
<td>End Time – Format: HHMMSS</td>
</tr>
</tbody>
</table>

#### Dates

- `<date-reg-ex>`: YYYYMMDD | (reg ex with .[^ $ and numbers to describe a date with format YYYYMMDD)
- `<reldays>`: D-XX | D-XX..D-XX (where XX is number of days relative to current date when filter runs)
- `<rel-day-of-week>`: W-XX (Where XX is number of working days relative to current date when filter runs not counting week-end ie: D-01 on monday is Friday)

#### Filter

Set the Time Series on which the Aggregation will be defined

### 4.10 Compression Configuration

Data compression rules can be defined to compress data. Rules can be prioritized, as needed using the UP/DOWN arrows.

Synchro stores raw data it receives in binary sparse files in the directory `<SynchroServer/data/metrics>`. These sparse files can be change any time. Data can be inserted “out of order” form the time perspective.

Compression of data runs daily to limit the amount of disk used by the raw sparse file. Once compressed, the data becomes read only (cannot be changed) – these compressed files are stored in the directory `<SynchroServer/data/metrics.gz>`.

**FOR LARGE SYNCHRO CONFIGURATIONS**

Synchro does a significant amount of small IO in numerous sparse files that take a relatively small amount of disk space – it is a recommended practice to set these files on fast disks (SSD drives are perfect for this job).

The compressed data (read only) can take significantly more space (depending retention policy) and is less IO intensive – it is a recommended practice to set this data on slower large disks infrastructure. The Synchro administrators can easily change the location of the directories using symbolic links.
<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Days</td>
<td>Number of days after which the data will be compressed and become read only</td>
</tr>
<tr>
<td>Dates to keep</td>
<td>Exception to the rule can be made in the following format:</td>
</tr>
<tr>
<td></td>
<td>&lt;date-reg-ex&gt;</td>
</tr>
<tr>
<td></td>
<td>date-reg-ex := YYYYMMDD</td>
</tr>
<tr>
<td></td>
<td>reldays := D-XX</td>
</tr>
<tr>
<td></td>
<td>rel-day-of-week := W-XX (Where XX is number of working days relative to current date when filter runs not counting week-end ie: D-01 on Monday is Friday)</td>
</tr>
<tr>
<td>Filter</td>
<td>Set the Time Series on which the rule will be defined</td>
</tr>
</tbody>
</table>

### 4.11 Retention Configuration

Data retention rules can be defined to destroy unused data. Rules can be prioritized as needed using the UP/DOWN arrows.

<table>
<thead>
<tr>
<th>Function Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Days</td>
<td>Number of days after which the data will be deleted</td>
</tr>
<tr>
<td>Dates to keep</td>
<td>Exception to the rule can be made in the following format:</td>
</tr>
<tr>
<td></td>
<td>&lt;date-reg-ex&gt;</td>
</tr>
<tr>
<td></td>
<td>date-reg-ex := YYYYMMDD</td>
</tr>
<tr>
<td></td>
<td>reldays := D-XX</td>
</tr>
<tr>
<td></td>
<td>rel-day-of-week := W-XX (Where XX is number of working days relative to current date when filter runs not counting week-end ie: D-01 on Monday is Friday)</td>
</tr>
<tr>
<td>Filter</td>
<td>Set the Time Series on which the rule will be defined</td>
</tr>
</tbody>
</table>

### 4.12 Client Configuration

Sets the configuration of the SynchroClient running on the SynchroServer. See APPENDIX A – to learn more about the SynchroClients configuration.

### 4.13 Dropfrom Configuration

Sets a List of hosts or IPs (one per line) from which the SynchroServer will not accept connection on the STAT port (stop receiving Time Series values).

If needed, it is also possible to close the STAT port all together so the server becomes “read only” – leaving only the HTTP port open for SyncroWEB interfaces.
4.14 Memory Trigger Exception Configuration

The SynchroServer can automatically switch to protect mode in condition where memory available is too low. This condition is detected using 2 indicators:

- Percentage of free memory threshold
- Duration in ms of the Java Garbage Collect process

These parameter values can be set in the SynchroServer configuration menu. Once one of these thresholds is reached, SynchroServer switches into memory protect mode and disconnects all clients and probes leaving only the HTTP port open.

The Memory Trigger Exception Configuration List is composed of a set of hostnames of IPs (One that will not be disconnected by SynchroServer when switched into protect mode).

4.15 Time Series Mapping Configuration

The SynchroServer can create new Time Series based of existing ones by grouping them together. The Time Series mapping configuration defines a list of rules that tells the SynchroServer how to group the Time Series together. This scheme is also a simple way to change the name of a Time Series as needed.

The syntax is:

Component1_Type1_Metric1_Target1=ComponentA_TypeA_MetricA_TargetA
Component2_Type2_Metric2_Target2=ComponentA_TypeA_MetricA_TargetA

4.16 Value Mapping Configuration

The SynchroServer can receive events using the trivial Text based TCP protocol – See APPENDIX B for more details. These events can be represented by a string characters like “Open”, “Close” that are converted by the SynchroServer into numerical values to be processed as such like any other Time Series.

SC01 [m]stat server.time type_metric_target=~sometextvalue

The Value Mapping configuration defines a set of rules that tells the SynchroServer how to handle these strings of characters by associating them with a numerical value.

Format:

sometextvalue=[number]

4.17 Presets Configuration

Window Controls are available when the mouse pointer hovers on the upper left portion of any Data Window (below the Title region). The Preset configuration allows the modification of some of these menus as needed.
Appendix A. SynchroClients

SynchroClients are used to collect technical metrics from computers. The metrics are collected every seconds and includes the following technical measurements when available:

<table>
<thead>
<tr>
<th>Target</th>
<th>Type</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Level</td>
<td>HOST</td>
<td>CPU, MEM, SWAP, PLIST, RUNQ, NICE, STEAL, SYSTEM, USER, WAIT, (Globally and per core when available)</td>
</tr>
<tr>
<td>All Processes running on the host</td>
<td>PROC</td>
<td>CPU, MEMPCT, RSSPCT, RSSMBY, MEMMBY</td>
</tr>
<tr>
<td>All network interface attached to the host</td>
<td>NET</td>
<td>RXPCKSEC, TXPCKSEC, TXKBYSEC, RXKBYSEC</td>
</tr>
<tr>
<td>All disk attached to the host</td>
<td>DISK</td>
<td>TPS, WSEC, RSEC, KBYWSEC, KBYRSEC, IOWAIT, QLEN, AWTIME, IOMSEC, STIME</td>
</tr>
<tr>
<td>All File Systems Mounted</td>
<td>FS</td>
<td>AVAIL-GB, SIZE-GB, USAGE-PCT, USED-GB</td>
</tr>
</tbody>
</table>

A.1 SynchroLinux

It is possible to throttle the SynchroLinux SynchroClient. There is a configuration file present on each SynchroLinux clients:

```
synchro@affa2:$ cat SynchroLinux/conf/client.ini
CPU=yes
DISK=yes
MEM=yes
NET=yes
PROC=yes
FS=yes
PROC-CPU-THRESHOLD=0.01
PROC-MEM-THRESHOLD=0.011
PROC-RSS-THRESHOLD=0.01
```

The Boolean switches CPU (CPU usage and other metrics at host level), DISK (per disks subsystem metrics), MEM (memory usage at the host level), NET (network interfaces metrics), PROC (processes metrics), FS (file system metrics) are used to turn on and off the associated metric collection by the SynchroClient.

If the PROC switch is on ("yes") (Process metrics collection is activated), you can use the PROC-XXX-THRESHOLD to fix the level of sensibility to processes (expressed in percentage). For example a value PROC-CPU-THRESHOLD=10 will report statistics only for processes that use at least 10% of one core of CPU (thus removing some of processes and keeping only "important processes").

A.2 SynchroWindows

It is possible to deploy the SynchroWindows without having the installation script displaying the notification window. For this just a "--quiet" on the command line.
Appendix B. Application Integration

This section presents information on how to integrate Synchro with other applications.

B.1 Trivial TCP Text Protocol

The following is a description of the protocol to use.

Here is the client protocol to integrate an application to Synchro.

Protocol Overview:

- The protocol is text based.
- Each message is one line of text.
- Messages must be cleaned from any of these characters: [\]~\@ and not printables
- Each field of the message is separated by a tab character.
- Each message starts with a special header.
- You can test the protocol using: telnet <server> 9123
- When the client connects, it must identify itself using the init message. The server will answer with the conf message.
- When init is done, the client can now start sending stat messages.
- Sending dumpconf message is completely optional.

Here is the description of the different messages:

*Client init message:* The /*init*/ message is mandatory and must be the first message sent to the server. This message is used to identify the client at the end of the connection.

Header: SC01
Command: /*init*/
Parameter: <client name>
End-of-message: \n
For example, this would be the C string for the *init* message if the client name is Computer1:

```c
sprintf( s, "SC01\ntinit\ts\n", "Computer1" );
```

*Client stat message:* The /*stat*/ message is used to send the metric values to the server.

Header: SC01
Command: /*stat*/
Parameter1: < YYYYMMDD.HHMMSS >
Parameter2: <metric value>
...
Parameter<n>: <metric value>

This would be the C string for the /*stat*/ message:
The date format is:

```c
sprintf(s, "%04d%02d%02d.%02d%02d%02d", year, month, day, hour, minute, second);
```

Alternatively, the user can select to use the server time instead of the client. To use the server time replace “YYYYMMDD.HHMMSS” by “server.time” string.

For example: the metric value format is (PROC is the type, CPU is the metric, perl is the target, 32.99 is the actual cpu time of the perl process):

```c
sprintf(s, "%s_%s_%s=%8.2f", "PROC", "CPU", "perl", 32.99);
```

You can append many metrics to the message, as long as they are all from the same date/time. Each metric value must be separated by a ‘\t’.

```c
sprintf(s, "SC01\tstat\t20080101.235559\tPROC_CPU_perl=32.99\n", );
```

**Client dumpconf message:** The /*dumpconf*/ message is not mandatory and is used to send back to the server some configuration information about the client.

- **Header:** SC01
- **Command:** /*dumpconf*/
- **Parameter 1:** <filename>
- **Parameter 2:** <line1>
- **Parameter 3:** <line2>
- ... 
- **Parameter <n>:** <line <n>>

This would be the C string for a /*dumpconf*/ message:

```c
sprintf(s, "SC01\tdumpconf\tfile1\tline1\tline2\n");
```

**Server conf message:** The server /*conf*/ message is used by the server to tell the client about some configuration parameters.

- **Header:** SS01
- **Command:** /*conf*/
- **Parameter 1:** Stat=n
- **Parameter 2:** DumpConf=1
- **Parameter 3:** <line 1 of client conf file>
- **Parameter 4:** <line 2 of client conf file>
- **Parameter <n>:** <line <n> of client conf file>

Stat=n is the number of seconds the server is expecting stat message from the client. Currently used by the Windows and Unix client for the general metric values. It is ignored by the ExternalSource plugins.

DumpConf=0 or DumpConf=1 is the server asking for basic configuration information which the client should send when the first ever connection is made. This is usually to help diagnose and identify client setup and host machine. It is not mandatory for the client to answer the DumpConf request.

**Telnet session sample:**
Appl  ication Integration

B.2 SynchroTail

SynchroTail is a very useful tool to read application logs. It understands log file patterns, and can change files automatically when new files are created. SynchroTail can also be restarted to continue reading from the file it was reading when it was stopped.

Here are some of the parameters of SynchroTail -

```
java -jar SynchroTail.jar "FilePattern=<regex>" [FirstFile=<filename>] [SortBy=time|filename] [FileTimeout=<nb seconds>] [EOFTimeout=<nb seconds>] [BaseDir=<dirname to search>] [ZTail=yes|no]
```

All parameters are optional except for FilePattern.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FilePattern</td>
<td>Needs to be a valid regular expression (not shell match).</td>
</tr>
<tr>
<td>FirstFile</td>
<td>Allow skipping of files (as in a restart mode) – files will be skipped until the FirstFile.</td>
</tr>
<tr>
<td>SortBy</td>
<td>“time” or “filename” (case sensitive)</td>
</tr>
<tr>
<td>FileTimeout</td>
<td>Is the number of seconds that the program will wait for the EOF before concluding that the file is completed. Default is five (5) seconds.</td>
</tr>
<tr>
<td>EOFTimeout</td>
<td>Is the number of seconds that the program will wait for new files before concluding that no additional files are to be processed. Default is five (5) seconds. The program waits FileTimeout on a particular file, and then waits EOFTimeout to see if other files are to be processed. If no file appears, the program stops.</td>
</tr>
<tr>
<td>RestartFileName=filename</td>
<td>Is the name of the file where SynchroTail will store the information needed to restart.</td>
</tr>
<tr>
<td>Restart</td>
<td>Is the parameter used to request SynchroTail to restart at the offset in the files (where it was when it was stopped).</td>
</tr>
</tbody>
</table>
Other parameters to control flow:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lps=XXX</code></td>
<td>Lines Per Second is the maximum number of lines per second that can be read in a file by SynchroTail.</td>
</tr>
<tr>
<td><code>delimiter=Char</code></td>
<td>Is the character to be used as a delimiter in place of end-of-line.</td>
</tr>
<tr>
<td><code>mbps=XXX</code></td>
<td>Mega Byte Per Second is the maximum number of bytes per second that can be read in a file by SynchroTail.</td>
</tr>
</tbody>
</table>

B.3 Integration Examples

This section presents example on how to integrate new ??????

Example – Perl Fusion IO Monitor

In this example, the user wants to monitor the number of FreeLebs in FusionIO drives. The script uses the server time instead of the client time.

```perl
#!/usr/bin/perl -w
BEGIN
{push @INC, '/home/synchro/FusionIOMonitor/';}

use Net::Telnet;

$Server="SynchroServerHostName";
$Port=9123;

$ServerID = new Net::Telnet (Telnetmode => 0,
Output_record_separator => "");
$ServerID->open(Host => $Server,Port => $Port);

$HOST=`hostname`;chomp($HOST);
$ServerID->print("SC01\ntinit\nFIOMonitor\n");
$Line = $ServerID->getline;
print ($Line);
while (1) {
$X=`cat /proc/fusion/fio/fioa/data/free_lebs`;chomp($X);
$Line1= "SC01\ntmstat $Time\nFIOMonitor\nFIOL_FreeLebs\nHOST=fios-$X\nt";
ServerID->print("$Line1\n");
sleep (1);
}
```

Example – Bash Latency Monitor

In this example, the user wants to monitor the latency between a Linux server and some hosts list contained in a file (HostList.txt). There is one host name per line. The user uses the host where the script is running time instead of the SynchroServer's time.

FILE CONTAINING HOST LIST:

```
synchro@affa2:~$ cat HostList.txt
HostName1
HostName2
```

SCRIPT:

```bash
#!/bin/bash
while (true); do
  if [ "$Init" = "1" ]; then
    Time=`date +%Y%m%d.%H%M%S`;
    echo -en "$SC01 mstat $Time\n";
    for Host in `cat HostList.txt`; do
      Latency=`ping -c 1 affa8|grep "time=" |cut -d= -f4|cut -d' ' -f1,1`;
  fi
  break
```

```bash
Example – Bash Latency Monitor

In this example, the user wants to monitor the latency between a Linux server and some hosts list contained in a file (HostList.txt). There is one host name per line. The user uses the host where the script is running time instead of the SynchroServer's time.

FILE CONTAINING HOST LIST:

```
synchro@affa2:~$ cat HostList.txt
HostName1
HostName2
```

SCRIPT:

```bash
#!/bin/bash
while (true); do
  if [ "$Init" = "1" ]; then
    Time=`date +%Y%m%d.%H%M%S`;
    echo -en "$SC01 mstat $Time\n";
    for Host in `cat HostList.txt`; do
      Latency=`ping -c 1 affa8|grep "time=" |cut -d= -f4|cut -d' ' -f1,1`;
  fi
  break
```
```
Example – Application Monitor using SynchroTail

The following is an example of a log file adapter using the SynchroTail application. SynchroTail monitors the end of files (EOF), using file pattern, as they are created by the application and prints out on STDOUT the data added to the files in real time (it tails application log files).

In this example, a simulator creates an hourly log file representing transaction made by clients on products. The format of the log file name is as follow:

LogFile-YYYYMMDD-HH.log where

• YYYY is the Year, MM is the Month, DD is the Day
• HH is the Hour.

The format of the log is

• Date (YYYYMMDD);
• Hour (HHMMSS);
• ClientID (Integer);
• ProductID (Integer);
• Qty (Integer).

EXAMPLE OF FILE CONTENT

```

host:~/TailExample/bin$ head ../data/LogFile-20120927-01.log
20120927;010000;00001;00008;00677
20120927;010000;00002;00007;00956
20120927;010000;00001;00003;0021
20120927;010000;00005;00000;00253

```

See the simulator script ExampleFileCreation.pl below.

The perl script (Injector-Tail-Example-01.pl) that parses the data created by the application (a simulator in this case) uses SynchroTail and pipe the data it reads as the data becomes available.

The outcome is a second by second aggregation so the transaction per Client and per Product.

USAGE EXAMPLE

```

host:~/TailExample/bin$ ./Injector-Tail-Example-01.pl -component TailExample -server 192.168.0.98 -file LogFile-20120928\.*log -dir ../data

```

SCRIPT

```

host:~/TailExample/bin$ cat Injector-Tail-Example-01.pl
#!/usr/bin/perl -w

use English;
use Getopt::Long;
```
use IO::Handle;
use Net::Telnet ();

my $Version="Version 0.0.3";
my %opts = (
  "help" => \$Help,
  "version" => \$ShowVersion,
  "Dir=s" => \$Dir,
  "Date=s" => \$Date,
  "Server=s" => \$Server,
  "Port=s" => \$Port,
  "Component=s" => \$Component,
  "Java=s" => \$Java,
  "LogDir=s" => \$LogDir,
  "LogFile=s" => \$LogFile,
  "File=s" => \$File,
  "Debug" => \$Debug,
  "Restart" => \$Restart,
  "SyncroTail=s" => \$SynchroTail,
);

GetOptions(%opts); &usage() if ($Help); die "$Version
n" if $ShowVersion;
sub Print;
sub Execute;
sub ProcessData;
die ("Need a Component name, use the -Component switch") unless defined($Component);
die ("Need a Server name to connect to, use the -Server switch") unless defined($Server);
$SynchroTail=":/SynchroTail/lib/SynchroTail.jar" unless defined ($SynchroTail);
die "Cannot find SynchroTail.jar, use the -SynchroTail switch (default:$SynchroTail)" unless (-e $SynchroTail);
$Java="java" unless defined ($Java);
#die "Cannot find java, use the -java switch (default:$Java)" unless (-e $Java);
die("Need file Pattern to proceed, use the -file switch.");unless defined ($File);
$Dir="/" unless defined ($Dir);
$DateCmd="date" unless defined ($DateCmd);
$Date=`$DateCmd +%Y%m%d` unless defined ($Date);chomp($Date);
$LogDir="/log/" unless defined ($LogDir);
$LogFile="$LogDir/TailExample.$Date.log" unless defined $LogFile;
open ( LOGFILE, ">$LogFile") || die "Cannot open file : $LogFile"

$Port=9123 unless defined ($Port);
$ServerID = new Net::Telnet (Errmode="return",Timeout => 10, Telnetmode => 0,
Output_record_separator => "="); $ServerID->open(Host => $Server,Port => $Port);
my $OldTime=""; my $GlobalRate=0; my $MaxRate=30000000; $Delay=0.01; #100
sub MatchExpression;
sub PrintData;
sub PrintTarget;
sub PrintMetricString;
$TempString="Java -jar $SynchroTail lps=50000000 synchroserver=$Server serverport=$Port
logfile=$LogDir/SynchroTail.log sortby=filename basedir=$Dir filepattern="$File"
RestartFileName=restart.ini RestartAllowOverlap=yes|i)
$TempString="Java -jar $SynchroTail restart.ini|" if defined ($Restart);
printf (LOGFILE "$TempString"") if defined($Debug);
only ( DATAFILE, "$TempString") || die "Cannot open $TempString";
while ($Line=<DATAFILE>){
  print ($MyDate, $Time,$Component,"Client","Quantity","ALL",$Qty) unless
  (defined($NoSend));
  print ($MyDate, $Time,$Component,"Client","Quantity","ALL",$Qty) unless
  (defined($NoSend));
Print($MyDate, $Time,$Component,"Product","Quantity","$ProductID","$Qty) unless (defined($NoSend));
Print($MyDate, $Time,$Component,"Product","Quantity","ALL","$Qty) unless (defined($NoSend));
}

sub Print {
    my ($MyDate, $MyTime,$Component,$Type,$Metric,$Target,$Value)=@_;
    $TmpStr=$MyDate."\".$MyTime;
    if (!($MyTime eq $OldTime)){
        Send("SC01\tinit\t$Component\n") if ($OldTime eq "");
        printf (LOGFILE "#SENDING SC01\tinit\t$Component\n")if defined ($Debug);
        $MetricRate=0 if ($OldTime eq "");
        Send("\n") unless ($OldTime eq "");
        select(undef, undef, undef, $Delay);
        Send("SC01\tmstat\t$TmpStr\t\n");
        printf (LOGFILE "#SENDING SC01\tmstat\t$TmpStr\t\n")if defined ($Debug);
        $OldTime=$MyTime;
        $GlobalRate=0;
    }
    $MetricRate++;
    if ($GlobalRate > $MaxRate){
        Send("\n");
        printf (LOGFILE "#SENDING \n")if defined ($Debug);
        select(undef, undef, undef, $Delay);
        Send("SC01\tmstat\t$TmpStr\t\n");
        printf (LOGFILE "#SENDING SC01\tmstat\t$TmpStr\t\n")if defined ($Debug);
        $GlobalRate=0;
    }
    $GlobalRate++;
    my $MyTmpTarget=substr($Target,0,30);
    $MyTmpTarget=~s/\[\]=/\-=/g;
    $TmpStr=$Component."\".$Type."\".$Metric."\".$MyTmpTarget."\".$Value;
    Send("$TmpStr\t");
    printf (LOGFILE "#SENDING $TmpStr\t\n")if defined ($Debug);
}

sub Send{
    my ($SendTmpString)=@_;  
    $ServerID->print("$SendTmpString");
    $Status=$ServerID->errmsg;
    while (! ($Status eq "")){
        sleep(1);
        print ("CONNECTION ERROR -retrying...
");
        printf (LOGFILE "CONNECTION ERROR -retrying...\n");
        $ServerID->close(Host => $Server,Port => $Port);
        $ServerID->open(Host => $Server,Port => $Port);
        $ServerID->print("SC01\tinit\t$Component\n");
        $Status=$ServerID->errmsg;
        if ($Status eq ""){
            print ("CONNECTION OK\n");
            printf (LOGFILE "CONNECTION OK\n");
            $ServerID->print("$SendTmpString");
            $Status=$ServerID->errmsg;
        }
    }
}

Grouping of all clients and products activity also creates a second by second overall activity by clients and products id done by the expression below. Instead of having the script sending another transaction, this grouping could have also been be done by the SynchroServer using “Time Series Mapping”.

Print($MyDate, $Time,$Component,"Product","Quantity","ALL","$Qty) unless (defined($NoSend));
#!/usr/bin/perl -w
use Getopt::Long;

my %opts = (
    "Date=s"        => \$Date,
    "NBClient=s"    => \$NBClient,
    "NBProduct=s"   => \$NBProduct,
    "MaxQty=s"      => \$MaxQty,
    "Rate=s"        => \$Rate,
);
GetOptions(%opts);

>DateCmd="date" unless defined ($DateCmd);
>Date=`$DateCmd +%Y%m%d` unless defined ($Date);chomp($Date);

$NBClient=10 unless defined ($NBClient);
$NBProduct=100 unless defined ($NBProduct);
$MaxQty=1000 unless defined ($MaxQty);
$Rate=1 unless defined ($Rate);

for ($Hour=0;$Hour<24;$Hour+){
    $HourStr = sprintf("%02d%02d%02d", $Hour,$Min,$Sec);
    $ClientStr=sprintf("%05d", int(rand($NBClient)));
    $ProductStr=sprintf("%05d", int(rand($NBProduct)));
    $QtyStr=sprintf("%05d", int(rand($MaxQty)));
    printf(LOGFILE
$Date$Separator$HourStr$Separator$ClientStr$Separator$ProductStr$Separator$QtyStr
);}

./ExampleFileCreation.pl -rate 2 -nbclient 1 -nbprod 1000 -date 20120922
ls -tr *.log|head LogFile-20120922-00.log LogFile-20120922-01.log LogFile-20120922-02.log
head LogFile-20120922-02.log
20120922;020000;00000;00588;00190
20120922;020000;00000;00151;00259
20120922;020000;00000;00754;00003
20120922;020000;00000;00310;00796
20120922;020000;00000;00050;00588

#!/usr/bin/perl -w
use English;
use Getopt::Long;
use IO::Handle;
use Net::Telnet ();

my $Version="Version 0.0.3";

my %opts = (
    "help"        => \$Help,
    "version"     => \$ShowVersion,
    "dir=s"       => \$Dir,
    "date=s"      => \$Date,
    "server=s"    => \$Server,
    "port=s"      => \$Port,
"Component=s" => \$Component,
"Java=s" => \$Java,
"LogDir=s" => \$LogDir,
"LogFile=s" => \$LogFile,
"File=s" => \$File,
"Debug" => \$Debug,
"Restart" => \$Restart,
"SynchroTail=s" => \$SynchroTail,

GetOptions(%opts); &usage() if ($Help); die "$Version
n" if $ShowVersion;
sub Print;
sub Execute
sub ProcessData;
die("Need a Component name, use the -Component switch") unless defined($Component);
die("Need a Server name to connect to, use the -Server switch") unless defined($Server);
$SynchroTail=../.SynchroTail/lib/SynchroTail.jar" unless defined ($SynchroTail);
die("Cannot find SynchroTail.jar, use the -SynchroTail switch (default:$SynchroTail)" unless ($SynchroTail);
$Java="java" unless defined ($Java);
die("Cannot find java, use the -java switch (default:$Java)");
$LogFile="/log/" unless defined ($LogFile);
$File="file=" unless defined ($File);

$DateCmd="date" unless defined ($DateCmd);
$Date=`$DateCmd +%Y%m%d` unless defined ($Date);
chomp($Date);
$LogDir="log/" unless defined ($LogDir);

$Port=9123 unless defined ($Port);
$ServerID = new Net::Telnet (Errmode=>'return',Timeout => 10, Telnetmode => 0, Output_record_separator => ");

$SynchroTail="/SynchroTail/lib/SynchroTail.jar" unless defined ($SynchroTail);
$LogFile="/log/TailExample.$Date.log" unless defined ($LogFile);
open ( LOGFILE, ">$LogFile") || die "Cannot open file : $LogFile";

$MyDate="\$MyDate, $MyTime, $Component,"Client","Quantity", $ClientID,$Qty unless (defined($NoSend));
$MyDate, $Time, $Component, "Product", "Quantity", $ProductID,$Qty unless (defined($NoSend));
if (!$MyTime eq $OldTime)) {
    Send("SC01\tinit\t$Component\n") if ($OldTime eq "");
    printf (LOGFILE "#SENDING SC01\tinit\t$Component\n") if defined ($Debug);
    $MetricRate=0 if ($OldTime eq "");
    Send("\n") unless ($OldTime eq "");
    select(undef, undef, undef, $Delay);
    Send("SC01\tmstat\t$TmpStr\t\n");
    printf (LOGFILE "#SENDING SC01\tmstat\t$TmpStr\t\n") if defined ($Debug);
    $OldTime=$MyTime;
    $GlobalRate=0;
}
$MetricRate++; if ($GlobalRate > $MaxRate) {
    Send("\n");
    printf (LOGFILE "#SENDING \n") if defined ($Debug);
    select(undef, undef, undef, $Delay);
    Send("SC01\tmstat\t$TmpStr\t\n");
    printf (LOGFILE "#SENDING SC01\tmstat\t$TmpStr\t\n") if defined ($Debug);
    $GlobalRate=0;
}
$GlobalRate++; my $MyTmpTarget=substr($Target,0,30);
$MyTmpTarget=~s/\|\-
\+\_//g;
$TmpStr=$Component."._$.Type."._$.Metric."._$.MyTmpTarget."=".$Value;
Send("$TmpStr\t\n");
printf (LOGFILE "#SENDING $TmpStr\t\n") if defined ($Debug);
} sub Send{
    my ($SendTmpString)=@_;
    $ServerID->print("$SendTmpString");
    $Status=$ServerID->errmsg;
    while (! ($Status eq "")) {
        sleep(1);
        print ("CONNECTION ERROR - retrying...");
        printf (LOGFILE "CONNECTION ERROR - retrying...");
        $ServerID->close(Host => $Server, Port => $Port);
        $ServerID->open(Host => $Server, Port => $Port);
        $ServerID->print("SC01\tinit\t$Component\n");
        $Status=$ServerID->errmsg;
        if ($Status eq "") {
            print ("CONNECTION OK.");
            printf (LOGFILE "CONNECTION OK");
            $ServerID->print("$SendTmpString");
            $Status=$ServerID->errmsg;
        }
    }
}
Appendix C. SynchroServer Performance Tuning

To improve SynchroServer performance, use the following approaches:

C.1 Increase Disk Performance

This approach has proven itself to be the most effective approach for improving SynchroServer performance.

Synchro stores raw data it receives in binary sparse files in the directory <SynchroServer/data/metrics>. These sparse files can be changed any time. Data can be inserted “out of order” from the time perspective.

Compression of data, runs daily to limit the amount of disk used by the raw sparse file. Once compressed, the data becomes read only (cannot be changed). These compressed files are stored in the directory <SynchroServer/data/metrics.gz>.

FOR LARGE SYNCHRO CONFIGURATIONS

Synchro does a significant amount of small IO in numerous sparse files that take a relatively small amount of disk space.

Note: It is a recommended practice to set these files on fast disks (SSD drives are perfect for this job).

The compressed data (read only) can take significantly more space, depending on the retention policy and is less IO intensive.

Note: It is a recommended practice to set this data on slower large disks infrastructure.

Synchro Administrators can easily change the location of the directories using symbolic links.

C.2 Increase CPU Usage

SynchroServer is a multi-threaded Java application. By default there are four (4) threads that process incoming statistics and four (4) threads that process incoming queries on the data. It is possible to configure the number of these threads to increase performance.

Parameters available in “SynchroAdmin/Server/Internal Configuration” menu are:

<table>
<thead>
<tr>
<th>Merge Engine Count</th>
<th>Number of internal engines that process incoming Statistics from clients and probes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result Engine Count</td>
<td>Number of internal engines that process incoming HTTP queries from SynchroWEB interfaces.</td>
</tr>
</tbody>
</table>
C.3 Increase Memory Available

The amount of memory available to SynchroServer is fixed at start time. By default, SynchroServer will not use more than 60% of available physical memory available on the host running the server. This can be changed by fixing the parameter “Memory Allocation” in the “Synchro Admin/Server/Server” menu. (Unit Megabyte)