

Enterprise RTView

An InDetail paper by Bloor Research
Author : Philip Howard
Publish date : January 2008

Enterprise RTView is arguably the most versatile product that we have seen in the BAM/Operational BI space, as well as the most extensive... As the product becomes more widely known, we expect it to represent the benchmark against which other products are measured.

Philip Howard

Fast facts

Enterprise RTView is a platform for monitoring, aggregating and delivering information in real-time, regardless of whether that delivery is via alerts (and resulting automated actions), reports or dashboards. Environments where this product might be applicable include business activity monitoring (BAM), business process optimisation, complex event processing (CEP) and operational BI, amongst others.

Nevertheless, this description may be misleading. There are, after all, lots of products that will help you to build dashboards (for example). However, that's all that most of these products do. If you want to source information from multiple sources in real-time then you have to deploy a separate federated engine to bring all the data together, which you can then use to feed the dashboarding product. With RTView, this is not necessary because the product includes its own data federation capabilities. In addition, especially when compared to many existing BAM products, RTView is not at all prescriptive: in fact it is extremely flexible so that you can implement dashboards in whatever fashion suits your company and your users.

Key findings

In the opinion of Bloor Research, the following represent the key facts of which prospective users should be aware:

- There is no programming required to develop real-time dashboards and alerts using Enterprise RTView. Everything is based entirely on drag-and-drop or point-and-click dialogues, though there is the option to drop down into Java if required.
- RTView is extremely flexible. The developer can decide what the user will see and how he will see it, or he can leave some or many of those options to the user by allowing the user to input relevant parameters. Thus, for example, users could dynamically amend threshold levels for alerts.
- Alerts can be generated manually or they may be automated. Any number of subsequent actions of virtually any conceivable type can be generated from a single alert.
- Role-based security is provided.
- At the user level no installation is required, though you can run via a desktop if preferred.
- There are a very wide range of pre-built graphical, tabular and other objects that can be deployed within a dashboard that can be customised (without programming) more or less at will. You can drill down (with parameter passing) to lower levels of detail from any object.
- Over 60 analytical functions are provided out-of-the-box, and you can also develop your own via a GUI. Functions enable you to define compound thresholds, aggregations and other calculations before presenting information to the user.
- Not only does RTView enable real-time reporting of information, it also allows that real-time data to be combined with historical data for comparison, trending and similar purposes, including anomaly detection.

The bottom line

Enterprise RTView is arguably the most versatile product that we have seen in the BAM/Operational BI space, as well as the most extensive. It is not only comprehensive in terms of its platform, but it is also extremely flexible and easy-to-use both for the developer and the end-user. As the product becomes more widely known, we expect it to represent the benchmark against which other such products are measured.

Background information

SL Corporation (where SL stands for Peter Sherrill and Tom Lubinski, the company's founders) was established 24 years ago in 1983 as a provider of graphical software tools for high-end, mission-critical command and control systems. It has around 4,000 customers for these tools including NASA, the Kennedy Space Center, ABB and Honeywell.

With these tools, SL focused exclusively on selling its technology through OEMs and it was only with the introduction of Enterprise RTView in 2002 that the company started liaising directly with enterprise end users. Nevertheless, around 50% of the company's RTView customers (approximately 250 in total) have been the result of direct sales.

The company continues to have a strong partnership programme with RTView embedded within products such as BEA WebLogic Event Server and Progress Apama. The company also has substantial technology partnerships with TIBCO (which resells Enterprise RTView under their brand), Oracle, Streambase, Aleri, Coral8, Radware (InFlight) and IBM amongst others, as well as with various systems integrators.

SL Corporation web address: www.sl.com

Product availability

Enterprise RTView was first introduced in 2002 and the current release is version 4.8. There are, in fact, three flavours of RTView: Enterprise RTView, Enterprise RTView for TIBCO, and Enterprise RTView for Oracle Coherence (the flagship product of Tangosol before it was acquired by Oracle), with the latter two products providing specialised capabilities for the environments indicated. In this review we focus on the generic Enterprise RTView, but it is appropriate to make a few comments about the other two products.

Briefly, the Enterprise RTView for TIBCO application provides pre-built TIBCO monitoring frameworks that enable customised TIBCO monitoring applications, while Enterprise RTView for Oracle Coherence provides users with visual monitoring capabilities across an Oracle Coherence grid, showing grid status and effectiveness. An example of such a dashboard is illustrated in Figure 1.

Apart from Enterprise RTView, SL's other products are J/Developer, AX/Developer, Developer for .NET and C++/Developer, which are development tools specifically designed for the creation of high-performance real-time graphical interfaces. There are also J/Net, C++/Net and C++/Map Extension products, which provide extended functionality and industry-specific libraries. All four of the base development tools share a dynamic graphical editor called SL-GMSDraw, and the model files built in this environment can be reused across any of the four base tools. All of these offerings (which are collectively marketed as SL-GMS Development Tools) run under Windows, UNIX and OpenVMS.

Financial results

SL Corporation is a private company with offices in the United States and Japan. It is privately funded and the ongoing revenue stream from its command and control products provides the resources necessary for the forward development and marketing of RTView. The company has roughly 50 employees.

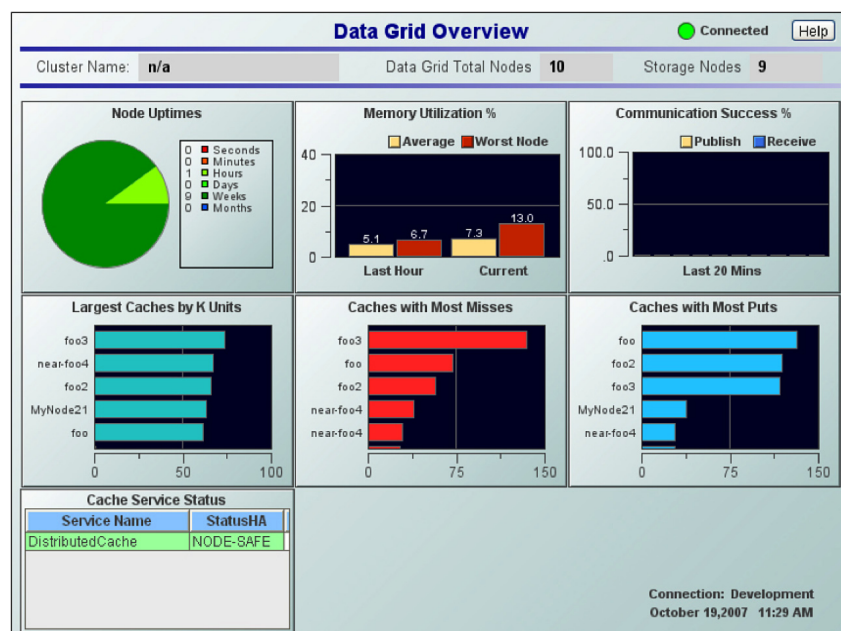


Figure 1: Dashboard view of Enterprise RTView for Oracle Coherence

Architecture

The various elements within Enterprise RTView are illustrated in Figure 2. We will spend the bulk of this review discussing the Server, Builder and the various information delivery options provided by RTView, but we do need to briefly discuss the other elements within this architecture.

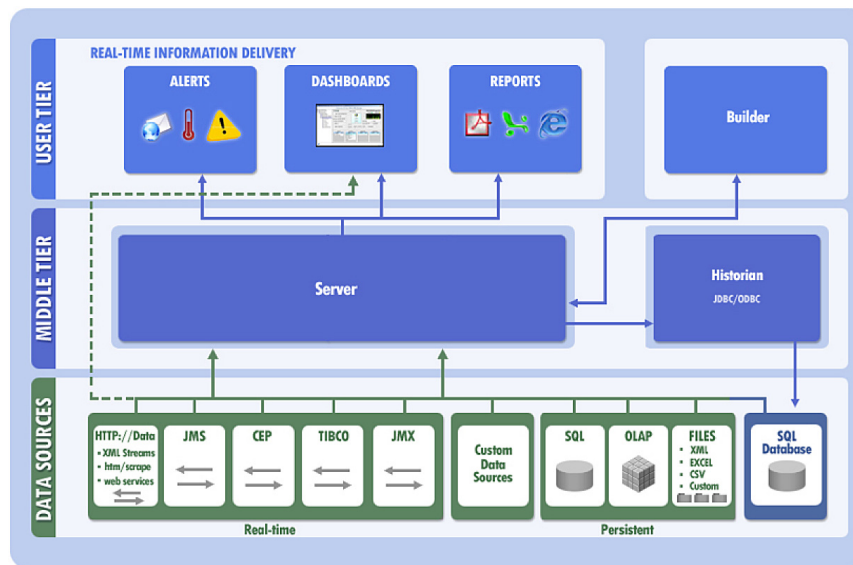


Figure 2: Enterprise RTView architecture

As far as the data sources listed, this diagram is not, in fact, complete. For example, there is support for IBM WebSphere MQ, and OSI Soft PI (which is relevant in the process control space); within the TIBCO category this includes TIBCO EMS (enterprise message service), Hawk and Rendezvous; and within the OLAP category there is support for OLE DB for OLAP (ODBO) and MDX as well as for Applix (now Cognos, soon to be IBM) TM1. For non-standard data sources you can convert relevant data into SL's standard XML format.

Note that RTView is not limited to working with real-time data sources but also supports access to existing, stored, data. This means that you can, for example, compare real-time information with historic data. Potentially, this could extend to comparisons with patterns of behaviour to detect anomalies that may represent threats (for example, fraudulent activity) or opportunities (such as early recognition of seasonal variations in retail sales).

The RTView Historian, which is used to store events and other information that you might want to analyse at a later date, or that you might want to be able to compare with future data (for example, using trend analysis or average upper-and-lower thresholds), can be driven from the relational database of your choice.

RTView can be deployed through a web browser (the software is based on the use of AJAX), via a thick client, or

via a JSR168-compliant portlet as required. There is no difference in functionality among these deployment options. At the back-end the server and development functions may be hosted by a variety of systems including both Windows and Linux. More specifically, RTView can be hosted on systems supporting JDK1.5 or 1.6-supported platforms including Windows Vista, XP, and 2003, Red Hat

Enterprise Linux 3 and 4, Sun Solaris 10, IBM AIX and Apple Macintosh. The product ships with the Tomcat application server but can be used with other comparable products if required.

The Server

The RTView Server is where the product really differentiates itself. This is where the federation of disparate data and data aggregation is handled, where the in-memory analytics engine resides, where alerts are activated and persisted, and where all distribution of data, graphics and reports is orchestrated. All of this will be covered in more detail below.

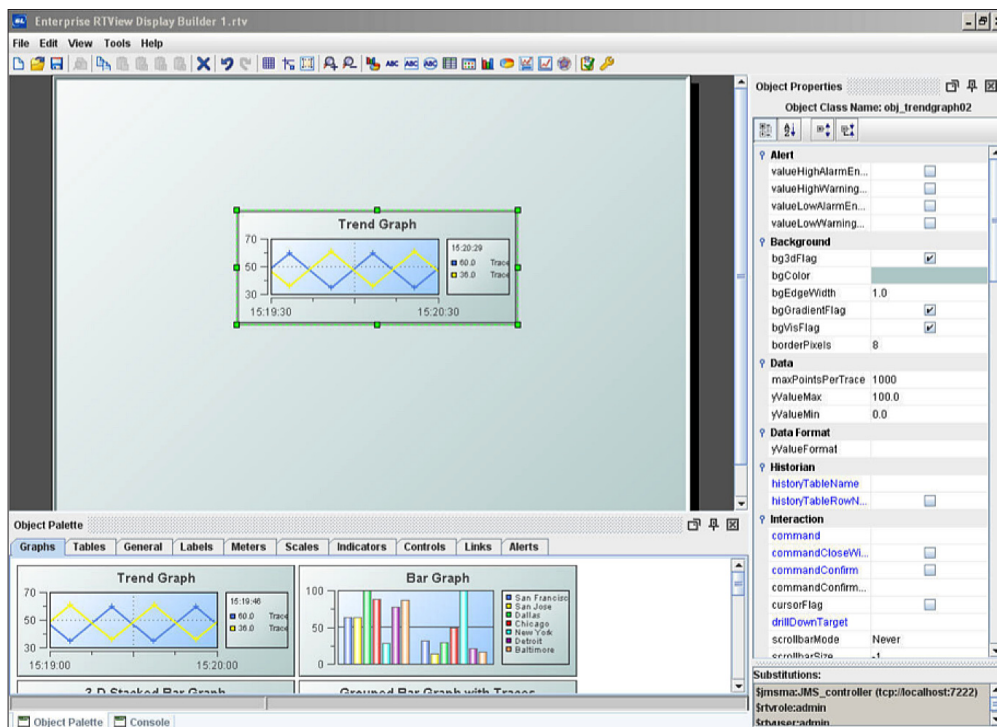
The Server supports users working on thin clients (you can

also run on the desktop) with updating on a specified basis (typically every 15 seconds) and, because of this servlet architecture, there is no reason why you could not deploy a dashboard (provided it was small enough) to a handheld device such as BlackBerry, if required. The Server also provides caching facilities so that data does not have to be retrieved from (persistent) source systems more than once.

The Builder

The Builder, a screenshot of which is illustrated in Figure 3, is where the developer designs the dashboards and other capabilities that will be deployed by users. However, in many respects the Builder acts as an extended user interface in the sense that data is displayed within in the Builder environment in just the same way as it does in the end-user environment. This is particularly useful for testing purposes, since the developer can see the effect of changing data on the dashboard as a part of the development process.

The first thing that you do with the Builder is to define the data sources that you are going to use and what specific data from those sources that you are interested in. This is fairly straightforward and is done in a manner that is similar to other integration tools. If you want to combine data from different sources prior to any processing then you can use one of the functions that are provided out-of-the-box, or you can define your own. These allow you to perform calculations, use Boolean logic and define "if... then... else"-type logic for compound threshold filtering, for example. That is, if you want to raise an alert only if two or

**Figure 3:** Builder screenshot

more conditions are fulfilled. Analytic functions, to a certain extent, duplicate SQL-type operations (as well as arithmetic and logical operations), but have additional real-time capabilities that you would not expect in SQL. For all pre-defined functions, there are built in pop-up boxes that explain the operation of each function to the developer.

Once the data you are going to use has been defined, you start working with the Object Palette as shown in Figure 3. Here you have a choice of various objects (see tabs at the bottom left), which can be dragged and dropped onto the palette. There are a wide range of graphs and charts that are supported and there are various display options for each graph including whether to display them in 2D or 3D, horizontally or vertically, using a waterfall (stepped increments) approach and so on. You can also combine methods: for example, by having a waterfall with a total bar. Tables are what their name suggests and, again, there are a number of options available. The same applies to Labels except that here it is worth noting that labels can be used as mouse-over flags or tool-tips. Meters, scales and indicators provide various continuous measurement options (dials, thermometers and so forth); controls provide widgets that can be used to interact with objects, and links are used for arrows and similar constructs that can be used in flow diagrams and the like. As mentioned, we will discuss alerts later.

As soon as you drag an object onto the palette, it becomes live and its properties are displayed, as shown to the right of the screenshot in Figure 3. Here the developer can change the attributes of an object (colour, style and so forth), attach the object to data (from the previously configured source or sources) or a function (which appears as if it were a data

source), apply filters and so on. In association with filters you can also apply, say, colour changes if the data breaks a filter threshold. Also, filters may be pre-defined by the developer or she can expose this to users with, say, a parameter, so that users can dynamically change thresholds, for example. Such parameterisation may also be applied to other object properties, as required.

Drill-down within RTView is treated as just another object property that can be defined against any object. In addition, when the user clicks on a data field to drill down from, then the value in the selected data field can be automatically reused as a variable within the next screen.

In so far as alerts are concerned these may be created manually by users or they can be generated automatically when values cross pre-defined thresholds. In the latter case, recall that such thresholds may be defined by analytic functions that incorporate multiple conditions, if required. For definition purposes, alerts are created using object properties by selecting the interaction option, where you can define as many actions as you like against a particular event. Such commands could range from executing some JavaScript or a stored procedure or SQL statement, to sending a message (via a queue, email, SMS or whatever), setting an SNMP trap, emitting a beep, opening a URL or running a DOS command or UNIX shell. Note that alerts can also be defined as a data source.

Finally, there are also facilities provided for users to automatically generate reports from their dashboards in .pdf, Excel, or HTML formats.

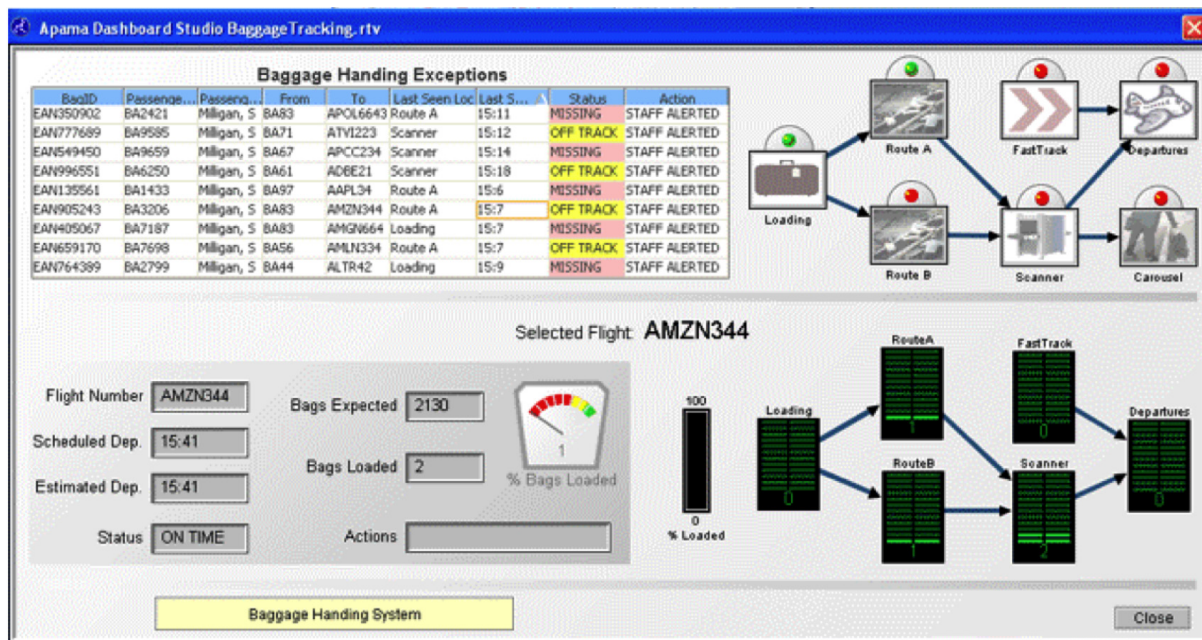


Figure 4: Example of BAM user dashboard

An example of a BAM (business activity monitoring) user dashboard developed via Enterprise RTView (as embedded within Progress Apama) for baggage handling is illustrated in Figure 4. In addition to the use of various objects and their properties that we have already discussed, note too the way that the company has made use of diagrams and pictures within the process flow illustrated. Such images can also be applied as a (faded) background within objects such as tables and charts.

Finally, one important point about the user environment is that if you are doing trend graphing of data, then, because it is real-time, this will only persist as long as your session remains live. If you close the dashboard and then re-open it later, your trend analysis will re-start. If you wish to be able to see trend information from previous sessions, RTView has the option of putting real-time data into a memory-resident cache. This is often used for storing trend data when it is not necessarily desirable to store it in a relational database via the Historian. The trade-off usually involves whether the historical data really needs to be persisted somewhere or if it is only immediately useful in a visual sense. This cache is also used to do high-speed, in-memory data analytics that may be too slow to perform by SQL queries.

Summary

Perhaps the most surprising thing about SL Corporation is how little known it is. Historically, the company has been entirely technically driven and it has achieved both success and longevity because of its product excellence rather than through any marketing. However, while that was fine for the highly technical market in which the company previously specialised, it is not enough in the BAM/Operational BI market that Enterprise RTView now addresses. While the company has had some significant success in this market through its partnerships with various other vendors, it would have continued to be a relatively unknown supplier had it not started to address the end-user market. We are therefore pleased to see that the company is growing its direct sales and we similarly welcome the company's new emphasis on marketing.

In our opinion, Enterprise RTView would be recognised as a market leader for BAM or Operational BI if it were better known. The company's increasing focus on marketing should help to build this awareness.

Bloor Research overview

Bloor Research has spent the last decade developing what is recognised as Europe's leading independent IT research organisation. With its core research activities underpinning a range of services, from research and consulting to events and publishing, Bloor Research is committed to turning knowledge into client value across all of its products and engagements. Our objectives are:

- Save clients' time by providing comparison and analysis that is clear and succinct.
- Update clients' expertise, enabling them to have a clear understanding of IT issues and facts and validate existing technology strategies.
- Bring an independent perspective, minimising the inherent risks of product selection and decision-making.
- Communicate our visionary perspective of the future of IT.

Founded in 1989, Bloor Research is one of the world's leading IT research, analysis and consultancy organisations—distributing research and analysis to IT user and vendor organisations throughout the world via online subscriptions, tailored research services and consultancy projects.

About the author



Philip Howard Research Director - Data

Philip started in the computer industry way back in 1973 and has variously worked as a systems analyst, programmer and salesperson, as well as in marketing and product management, for a variety of companies including GEC Marconi, GPT, Philips Data Systems, Raytheon and NCR.

After a quarter of a century of not being his own boss Philip set up what is now P3ST (Wordsmiths) Ltd in 1992 and his first client was Bloor Research (then ButlerBloor), with Philip working for the company as an associate analyst. His relationship with Bloor Research has continued since that time and he is now Research Director. His practice area encompasses anything to do with data and content and he has five further analysts working with him in this area. While maintaining an overview of the whole space Philip himself specialises in databases, data management, data integration, data quality, data federation, master data management, data governance and data warehousing. He also has an interest in event stream/complex event processing.

In addition to the numerous reports Philip has written on behalf of Bloor Research, Philip also contributes regularly to www.IT-Director.com and www.IT-Analysis.com and was previously the editor of both "Application Development News" and "Operating System News" on behalf of Cambridge Market Intelligence (CMI). He has also contributed to various magazines and published a number of reports published by companies such as CMI and The Financial Times.

Away from work, Philip's primary leisure activities are canal boats, skiing, playing Bridge (at which he is a Life Master) and walking the dog.

Copyright & disclaimer

This document is copyright © 2008 Bloor Research. No part of this publication may be reproduced by any method whatsoever without the prior consent of Bloor Research.

Due to the nature of this material, numerous hardware and software products have been mentioned by name. In the majority, if not all, of the cases, these product names are claimed as trademarks by the companies that manufacture the products. It is not Bloor Research's intent to claim these names or trademarks as our own. Likewise, company logos, graphics or screen shots have been reproduced with the consent of the owner and are subject to that owner's copyright.

Whilst every care has been taken in the preparation of this document to ensure that the information is correct, the publishers cannot accept responsibility for any errors or omissions.



Suite 4, Town Hall,
86 Watling Street East
TOWCESTER,
Northamptonshire,
NN12 6BS, United Kingdom

Tel: +44 (0)870 345 9911
Fax: +44 (0)870 345 9922
Web: www.bloor-research.com
email: info@bloor-research.com