



# TIBCO® General Interface

This whitepaper provides an introductory, high-level business and technical overview of TIBCO® General Interface, a powerful rich client software that delivers the look, speed and performance of desktop installed software, yet requires no end user installation. This whitepaper addresses the industry's need for rich client technology and TIBCO General Interface's unique architectural advantages that drive measurable benefits for the enterprise.



<http://www.tibco.com>

**Global Headquarters**

3303 Hillview Avenue  
Palo Alto, CA 94304  
Tel: +1 650-846-1000  
Toll Free: 1 800-420-8450  
Fax: +1 650-846-1005

©Copyright 2004, TIBCO Software Inc. All rights reserved. TIBCO, the TIBCO logo, TIBCO Software, TIBCO General Interface, TIBCO General Interface Framework, TIBCO General Interface Builder, and TIBCO Rendezvous are trademarks or registered trademarks of TIBCO Software Inc. in the United States and/or other countries. All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

## Table of Contents

---

<b>1</b>	<b>The Need for More Powerful Web Applications.....</b>	<b>3</b>
<b>2</b>	<b>How does TIBCO General Interface Framework Work?.....</b>	<b>3</b>
2.1	Framework Deployment and Initialization.....	3
2.2	Run-Time.....	3
<b>3</b>	<b>TIBCO General Interface Framework and MVC Client-Server Architecture</b>	<b>4</b>
<b>4</b>	<b>TIBCO General Interface Products .....</b>	<b>5</b>
4.1	TIBCO General Interface Framework .....	5
4.2	TIBCO General Interface Builder .....	5
<b>5</b>	<b>Integrating with Web Services and Messaging Technologies .....</b>	<b>6</b>
<b>6</b>	<b>Security .....</b>	<b>6</b>
<b>7</b>	<b>Benefits .....</b>	<b>6</b>
7.1	More Power for the End-user .....	6
7.2	Lower Development and Maintenance Costs .....	6
7.3	Lower Hardware Costs.....	7
<b>8</b>	<b>Conclusion.....</b>	<b>7</b>

## 1 The Need for More Powerful Web Applications

Despite years of innovation, applications accessible via Web browsers still do not deliver the robust features users expect from traditional desktop software. And while browsers have evolved to support everything from animation to applets, they remain mere display mechanisms for delivering pages of information, one click at a time. The big advances have come at the back end – in middleware technologies, databases, caching servers and development methodologies.

Accordingly, enterprises have been faced with limited front-end deployment options: Deploy server-generated HTML pages that are rendered by the ubiquitous thin-client Web browser, or deploy thick client applications and take on the challenges of installation conflicts, code updates, maintenance and support. Neither option is optimal. Web pages are low in cost, but they also offer low functionality. Thick clients have rich functionality, but are expensive. With TIBCO General Interface's rich client framework, enterprises can now have the best of both: applications that combine the low total cost of ownership and instant accessibility of Web deployed solutions with the rich functionality and performance of client-installed software.

Unlike Web page-based applications that require repeated page loads and are constrained by the limited components offered by HTML markup, applications powered by TIBCO General Interface deliver software-style functionality and rich components while running within a single HTML page within the browser and communicating with network services without requiring page reloads. No applets, plug-ins, active-X components, or installed client or server frameworks are required to run the TIBCO® General Interface Framework in any Web page and enable the instant delivery of browser-based applications that have the look, feel and performance of desktop installed software. Developers can author directly against the JavaScript APIs of the TIBCO General Interface Framework using their scripting environment of choice, or can use TIBCO® General Interface Builder to rapidly assemble graphical user interfaces and connect them to data and services. Now you can quickly create and easily deploy rich client functionality to power full screen applications or feature-rich modules embedded within Web pages and portal interfaces.

## 2 How does TIBCO General Interface Framework Work?

An application works according to two main processes.

### 2.1 Framework Deployment and Initialization

When an application is requested by linking to a simple URL, the TIBCO General Interface Framework, your application logic and your application component descriptions are passed to the browser as a series of small, cacheable JavaScript and XML files from any HTTP server of your choice. As the browser loads and interprets these files, it builds out the TIBCO General Interface Framework in memory. Then, calling upon the objects and methods of the TIBCO General Interface Framework, the browser instantiates the application model in memory and generates the user interface view leveraging the browser's existing rendering engine.

### 2.2 Run-Time

Once initialized, in response to business logic invoked programmatically or by end-user events, the application acquires data from the network threading out one or more synchronous or asynchronous messages. Services responding to the messages typically return data as XML or SOAP over HTTP/S though any structured ASCII data can be used. Data received is typically stored in the TIBCO General Interface Framework's client-side data cache for local use during the session. With a callback to a graphical user interface (GUI) component's paint method, the component updates the

view without requiring a screen refresh. The user interacts with the application and data locally without requiring additional interaction with the server except to acquire or send new data.

The result is a feature-rich, fast performing, application running in the browser including client-side data, logic, rich presentation functionality, and one or more server connections as shown in Figure 1 below.

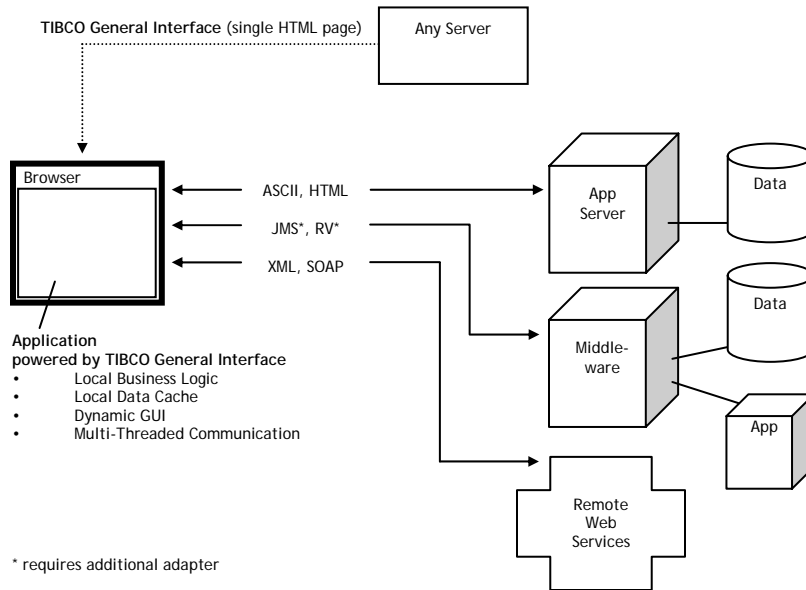


Figure 1. TIBCO General Interface Framework running in a browser

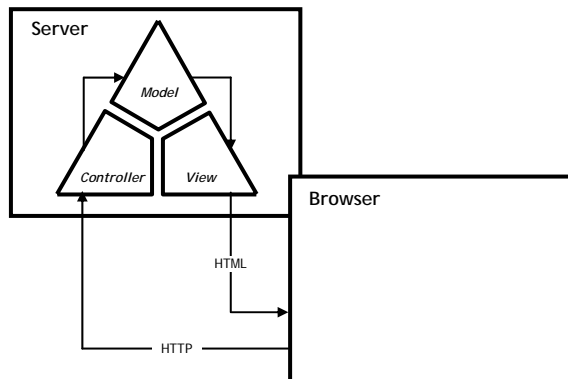
### 3 TIBCO General Interface Framework and MVC Client-Server Architecture

The TIBCO General Interface Framework uses model-view-controller (MVC) principles. This means that TIBCO General Interface components are broken down into three architectural elements, each playing a crucial role in how the component behaves.

In traditional server-based MVC architecture, all processes run on the server. Requests are sent to the Controller; Models are updated; the View returns a stream of HTML and data to the browser where it is rendered for the user. This results in system latency and reduced functional possibilities for the end-user. The HTML stream typically includes 80% presentation instructions and 20% data.

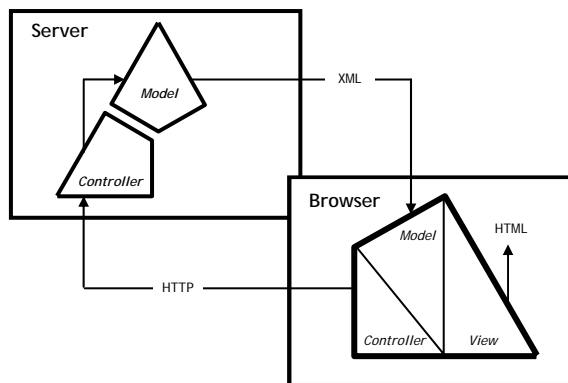
In contrast, TIBCO General Interface Framework enables the View portion of the MVC architecture to be instantly distributed to, and to run on, the client system in a Web browser. This approach improves application scalability by removing significant server processes; decreases latency by using 98% of the bandwidth to deliver data (instead of the 20% of bandwidth for data in the traditional model); and finally delivers full-featured application performance and rich GUI features to an unmodified Web browser (no plug-ins, no runtime environments, no extra software to install).

Figure 2 below illustrates this principle in action.



#### Server-Based MVC Architecture:

View processes run on the server returning streams of HTML and data to the browser. Browser is simply a HTML page rendering device.



#### TIBCO General Interface MVC Architecture:

View processes are distributed to the browser where they run on the client system. The distributed view employs a MVC Architecture of its own turning the browser into a rich Internet client environment. Client-side objects generate the HTML. Fewer processes need to run on the server. Bandwidth is optimized for data. Application performance and scalability are significantly enhanced.

Figure 2. Server-Based MVC vs. TIBCO General Interface MVC

## 4 TIBCO General Interface Products

### 4.1 TIBCO General Interface Framework

TIBCO General Interface Framework is the software that provides a deployment-time framework that instantly adds rich client capabilities to a standard browser. The TIBCO General Interface Framework provides a robust object-oriented JavaScript API consisting of multiple class objects organized into several packages: GUI, Data, Events, Communications, and System.

### 4.2 TIBCO General Interface Builder

You can use any script-based integrated development environment (IDE) to author applications that call upon the APIs of the TIBCO General Interface Framework. TIBCO also offers a powerful visual authoring environment optimized for rapidly creating TIBCO General Interface applications and components. With TIBCO® General Interface Builder developers can rapidly assemble application interfaces; build connections to servlets, XML data sources and Web services; manage the client-side XML data cache; and author the business logic that orchestrates the application. TIBCO General Interface Builder includes step-through debugging, integrated API documentation, WYSIWYG layout and extensible component libraries.

For detailed information please contact us.

## 5 Integrating with XML sources, Web Services and Messaging Technologies

TIBCO General Interface's communications classes enable TIBCO General Interface applications to communicate with any HTTP/S accessible sources of data. Data usually is provided in the form of XML or SOAP, though any structured ASCII data is valid.

The robust communication classes support HTTP GET, HTTP POST, XML, XML RPC, SOAP, and other compatible communication standards. Accordingly, TIBCO General Interface is agnostic to server environments from an integration point of view. Any server capable of accepting and receiving HTTP GET/POST, or XML/Web service structured data can communicate with TIBCO General Interface Framework.

A J2EE servlet can provide an XML interface into a sophisticated back-end application. A SOAP or .NET service can easily communicate with a TIBCO General Interface application. TIBCO General Interface applications can communicate with XML based middleware, messaging technologies (JMS, TIBCO Rendezvous®) or Enterprise Application Integration (EAI) and Business Process Management (BPM) tiers. Any application with an XML or Web services API can be connected over HTTP/S.

## 6 Security

TIBCO General Interface Framework runs completely within the browser container and therefore inherits all the SSL communication, authentication, certificate support and other capabilities of the browser itself. If you feel secure in deploying HTML pages, then you can feel secure in deploying TIBCO General Interface Framework.

## 7 Benefits

The efficient Web-based deployment model combined with object-based architecture of TIBCO General Interface drives down the costs of developing, maintaining, and deploying applications while simultaneously providing feature-rich applications that boost end-user productivity, utilization patterns, and satisfaction.

### 7.1 More Power for the End-user

HTML is good for document-centric applications such as marketing Web sites, document management solutions, knowledge management systems, and aspects of portal applications. TIBCO General Interface Framework is the answer when solutions requirements call for more powerful software-like features. Quote configurators, dashboards, analytics, call center solutions, supply chain logistics, underwriter workstations, field operations solutions, real-time trading stations, and many other types of applications all require richer functionality. Now you can deliver these feature-rich solutions in the way businesses want to deploy them: instantly through the Web browser.

### 7.2 Lower Development and Maintenance Costs

The Aberdeen group estimates that using the techniques employed by TIBCO General Interface, development and maintenance costs can be **reduced 30% to 50%**. The re-utilization and componentization of code, plus the rich

\* Requires additional adapter

foundation libraries, enable applications to be created quickly and to be more easily maintained. In addition, TIBCO General Interface includes a visual development environment for creating, maintaining, and deploying TIBCO General Interface applications that significantly reduces solution authoring time.

Development of the application's front-end and back-end are independent. Therefore application clients and application servers can be created in parallel — accelerating time-to-market. Updates to the client require little or no modification at the server, and vice versa.

### 7.3 Lower Hardware Costs

Presentation processes are off-loaded to the workstation, freeing up as much as **50% of server resources**. Therefore fewer servers are necessary to achieve equivalent scale.

Significantly less bandwidth is necessary as up to **95% of the data stream** that was previously consumed by HTML is now available for data transmission. That means either less corporate bandwidth requirements, or faster performance at dial-up speeds.

Existing workstation CPUs of 350 mhz and 128 megabytes of RAM are more than sufficient to get great performance out of TIBCO General Interface. This means that the latest desktop or notebook hardware is not required to deploy TIBCO General Interface.

For more information please contact us about total cost of ownership analyses and return on investment case studies.

## 8 Conclusion

TIBCO General Interface's architecture lends itself to ultra efficient distribution and development, enabling businesses to do more with less. Applications powered by TIBCO General Interface deliver the rich software-like interfaces and components that users prefer while bringing all the benefits and lower costs of centralized application deployment and maintenance to the enterprise. And with its distributed computing model, enterprises get more yield from existing application, hardware, and network investments. End-users receive more powerful and productive applications, IT departments significantly reduce costs, and application developers get more done on smaller budgets in shorter development cycles.