

MAGIC SOFTWARE ENTERPRISES LTD.

---

Magic eDeveloper of the Magic eBusiness Platform

# **eDeveloper: Developing for the Web**



*Concepts and Considerations*

# Preface

## Two Major Technologies

eDeveloper 9.4 provides these two technologies that you can use to develop Web applications:

- Merge
- Browser Client

This document describes each technology and explains the factors you need to consider before you use them to develop applications for the Web.

## Matching the Technology to Your Application

Before deciding which technology to use to create the application, you should identify your application's needs.

### Application Characteristics

You should consider these factors: Massive data entry, sophisticated browsing capabilities, heavy loaded views, and complex flow logic.

### Deployment Considerations

There are several issues regarding Web application deployment that you need to consider, such as:

- What is the available network traffic rate between the servers and clients?
- Will the application only be accessed from within a LAN, such as an Intranet, or also from a WAN, such as the Internet?
- Do you know who the expected end users are?
- Is the end user's browser configuration known?

## eDeveloper: Developing for the Web

### Functionality Comparison:

The table below compares Merge with Browser Client technology.

	<b>Merge</b>	<b>Browser Client</b>
<b>Application Flow</b>	Replaces pages	Replaces data on the page
<b>Application Logic</b>	Server side only	Client and Server
<b>Browser add-on</b>	None	Browser Client Java Module
<b>Browser Compliancy</b>	All	Microsoft® Internet Explorer 5.5 or higher
<b>Code complexity</b>	High (scripts, multiple programs)	Low (same as C/S) Singular paradigm
<b>Context</b>	Has to be built by the developer	Managed automatically by the server
<b>Execution mode</b>	Page	Field
<b>Interaction</b>	Can be achieved by the developer using Page Submit and additional JavaScript code	Automatically managed using event handling
<b>Maintainability</b>	Medium	Rapid
<b>Network Bandwidth requirements</b>	Low*	Medium/High*
<b>Transmission Modules</b>	Presentation HTML	Presentation - HTML Logic - XML Data - XML

\*Depends on the level of application complexity and interaction.

# Merge Technology

## General

The Merge technology lets the developer create dynamic Web pages on the server side as a response to HTTP requests.

## Concept

Using a set of tokens that are embedded in a regular HTML/XML file, the Enterprise Server can merge any application data into the HTML file to produce the dynamic Web page.

Every request for a dynamic Web page activates a corresponding batch program. This program can receive data from the request, such as submit form variables and cookies, process the application data according to the request information, and process the application logic to produce the merged Web page result.

## Characteristics

### Page Mode Execution

The interaction between the browser and the server is usually manifested in the retrieval of new pages as a result of a request submitted by the previous page.

### Compliance with All Browsers

The developer can choose the HTML/XML version for the application's Web pages and any other use of client-side scripts and modules. The developer can decide on the level of Web browser compliance by choosing the HTML/XML version and additional modules.

### Application Logic

The logic for merging application data into standard HTML files is server-side logic. Any client-side logic required for the application can be integrated with the eDeveloper-generated dynamic Web pages in the form of client-side scripts, such as JavaScript or VB script, and client-side modules, such as ActiveX controls and Java applets.

### Controlling the Interaction

The fact that the application logic executed by eDeveloper is server-side only and that the developer determines the client-side logic, including the available

## eDeveloper: Developing for the Web

hyperlinks, provides full control of the level of interaction between the client and the server.

### **Context Management**

Any request handled by the Enterprise Server is handled independently with no correlation to previous requests submitted by the same application context for a given end user. This means that the application flow context management should be constructed and maintained by the developer.

## **Considerations**

The Merge technology is suited for lightweight interaction between the browser and the server. This technology is designed for applications that mainly receive whole pages on each request.

### **Network Throughput**

The number of interactions between the server and the client is relatively low and the amount of information passed from the browser to the server in every interaction, for example an HTML form's submitted information, is also low. However, the result page for every request may be large as the page always returns not just the new processed data but also the entire HTML portions that define the interface and design.

Given the fact that the volume of information passed from the client to the server, i.e. the uploaded data, is usually low, excluding file transfers, there is no need for a large throughput from the client to the server, i.e. the upload rate does not need to be great.

You should try to make your pages as light as possible or make sure that the end-user machines have sufficient download capabilities.

### **Client Machine Requirements**

Unless the developer chooses to enhance the pages with various objects, such as ActiveX controls and Java applets, no special requirements are set for the client machine.

### **Unknown Users**

The ability to create applications supported by all browsers lets you freely distribute a Merge-based application with no need to know the end-user machine specifications.

### **Client-side Scripting Skills**

If the Web application requires client-side logic, the developers need to acquire knowledge and skills in client-side scripting.

# Browser Client Technology

## General

The Browser Client technology lets the developer create HTML-based interactive pages that provide both server-side and client-side logic without the need for additional client-side modules, such as client-side scripts.

## Concept

A request result page generated using Browser Client technology is regarded as a fully-fledged interactive task.

During development, the developer can define the server-side logic, such as dataview settings, initialization, and other server-side computation. The developer can also define the client-side logic by handling client-side events using fields, modifications, and navigation.

Moreover, Browser Client technology provides automatic handling of basic interactions, such as submission of data modifications, transaction management, and data scrolling.

MSE provides a methodology document for creating optimized Browser Client applications. The methodology document is included with the eDeveloper installation.

## Characteristics

### **Server-side and Client-side Logic**

Browser Client technology lets the developer define both server-side logic and client-side logic using a single development paradigm.

### **Microsoft® Internet Explorer 5.5 Compatibility**

Browser Client applications can only be executed on browsers that are fully compatible with Microsoft® Internet Explorer 5.5 or higher.

## eDeveloper: Developing for the Web

### Browser Client Modules

The ability to tightly handle the Web page on the browser is enabled by automatically equipping the browser with engine modules that handle the page logic defined by the eDeveloper task. These modules are transmitted only once at the first request and they can be kept persistently on the client side avoiding the need to receive these modules whenever the application is accessed.

### Data and Logic Transmission

The activated program's logic and data set are embedded in the result page as an XML island. This XML section is parsed by the Browser Client modules, allowing the modules to handle the page according to the logic and data.

### Page Durability

With an application based on dynamic server-side pages, additional data and data-driven feedback is displayed on the browser as new retrieved pages. However, in a Browser Client application, fewer pages are created for a particular application flow because additional data and data-driven feedback is retrieved and handled by the existing viewed page.

### Controlling the Interaction

In most cases, the Browser Client modules automatically control the interaction between the client and server. By following the eDeveloper *Interactive Web Applications* methodology, the developer can affect the number of interactions required for the application.

### Context Management

The Browser Client technology provides the developer with automated and fully controlled context management. The server identifies each request of an already-activated application as the consecutive request of a known application context and serves the request within the latest context state.

Various resources, such as global values and memory tables, can be set in a context by one request and accessed by consecutive requests belonging to the same context.

## Considerations

Browser Client technology is suited for high-level interactivity, such as real-time applications that consist mainly of data-entry pages.

### Network Throughput

The number of interactions between the server and the client depends on the level of interaction and the logic defined by the developer. Compared to page-mode applications, the number of pages in a Browser Client application will be smaller. However, the volume of information passed from the server to the client and back will be greater.

## **eDeveloper: Developing for the Web**

The throughput required for a Browser Client application depends on the dataview size, the amount of logic, and the level of interaction. In general, the throughput required for a Browser Client application, in terms of the upload and download rates, would be greater than the throughput required for a Merge application.

Appendix A of this document provides a description of Web-based application functionalities and their effect on data transfer and the level of client and server interaction.

### **Client Machine Requirements**

Running a Browser Client application requires execution of the Browser Client modules on the browser. This means that the client machine should have sufficient memory resources.

### **Client-side Scripting Skills**

Using Browser Client technology requires no client-side scripting skills. However, developers with extensive knowledge of client-side scripting and general Web skills can still easily integrate scripts into applications.

# Combining the Two Technologies

## General

A Web application can use both technologies and combine them according to the application needs.

## Data Entry vs. Query Results

Many applications consist of both interactive pages that provide data-entry capabilities and pages that display query results and processed reports. You can use the Browser Client technology to produce high-level interactive pages for the data entry, and use the Merge technology to produce the query results and processed reports.

## Methodology

The MSE methodology document about *Interactive Web Applications* describes how you can combine these technologies.

# Future Development

## **General**

The methods for developing and deploying Web applications using eDeveloper are constantly evolving and are subject to constant optimization and improvement.

## **Transparent Integration of Technologies**

eDeveloper will provide smooth integration of the two Web-enabling technologies to produce even more efficient and usable Web applications.

## **Optimized Level of Interactions**

The implicit interaction between the client and the server will be optimized by minimizing the occasions when the client is required to turn to the server.

## **Optimized Data Transport**

The data transported in every client or server interaction will be reduced to make the required interactions lighter in volume.

## **Enhanced Functionality**

eDeveloper will provide enhanced functionalities to make Web application development even faster. New functionalities will be added, such as built-in Menu support and enhanced Table controls.

# Appendix A

This appendix describes the Browser Client application functionalities that require the client to turn to the server. The appendix also describes the elements that affect the size of the transported data.

## Server-side Activities

The browser requires server processing for these functionalities:

- ❑ Re-computation of linked records
- ❑ Re-computation of subforms
- ❑ Calling another program
- ❑ Closing a Browser program
- ❑ Transaction commit
- ❑ Transaction rollback
- ❑ Retrieval of additional data due to scrolling or refreshing the page
- ❑ Re-computation of data controls
- ❑ Evaluation of server-side expressions

## Transported Data

Whenever there is interaction between the client and server, data is passed from the client to the server and vice versa. Large portions of transported data may slow down the total execution of every roundtrip.

The most significant portions of data are transported when the events described below occur.

### Opening a Browser Client Task

When a browser task is opened, the initial page is constructed and contains:

- ❑ The HTML display
- ❑ Task Logic: The more complex the task and the more logic the task contains, the larger the logic portion of the page will be.

## eDeveloper: Developing for the Web

- **Initial Data:** The size of the initial data is roughly the product of the record length multiplied by the chunk size setting. The larger the record or chunk size, the greater the portion of the initial data will be.

### **Retrieval of Additional Data**

Retrieval of additional data due to scrolling or refreshing the page will return the data of a full chunk size, similar to the initial data passed on opening the task.

### **Updating the Dataview on the Server**

Whenever the client turns to the server for server-side processing, the client must update the dataview image on the server to match modifications on the client. For every such interaction, the current records of every opened browser task are transmitted back to the server. Maintaining large-size records on the browser and many open tasks will increase the size of data transmitted from the client to the server.