

# Porting Legacy Windows Applications to the Server and Web

## About TX Text Control .NET Server:

TX Text Control .NET Server is a fully programmable word processing engine for deployment in an ASP.NET server environment. It primarily has been designed to develop document automation applications.

<http://www.textcontrol.com>

A common trend in the development of business software is the centralization of key processes, such as document creation and document management.

There are many reasons to port old legacy Windows applications to the server:

- Applications are independent of the client's operating system
- End users do not have to install software on their machines
- Processes can be controlled centrally

For developers, however, porting applications to the server is anything but a simple task - they face a steep learning curve, especially those coming from classic client application programming background.

---

## Content

What are server-based applications? .....	2
Template-based document processing .....	2
Centralizing the document creation processes .....	2
ASP.NET: WebForms and code-behind .....	3
Editing document in the browser .....	3
Full WYSIWYG: Client-side controls .....	4
Editing text blocks .....	4
Conclusion .....	5
Feasibility Discussion .....	5



# What are server-based applications?

The aim of this paper is to offer interested readers an overview of the challenges and problems developers may face when porting their applications to the server.

In particular, this paper examines word and document processing applications and provides suggestions on how to realize word processing processes server side.

Generally, all kind of business applications have to create documents: invoices, quotations, reminders or similar business correspondences. Typically, developers author applications that enable end users to modify text and to create print-ready documents that can be sent via e-mail (PDF) or fax.

Documents are normally created from templates that ship with the application. This makes it very difficult - if not impossible - to change the style of the final document or to replace the templates with one that reflect the enterprise's corporate identity.

Newer software architectures often offer the flexibility of creating and populating documents server side, thus centralizing critical business processes.

## What is a template?

A pre-designed document with merge fields that can be merged with data from various data sources like databases or XML files. Templates help developers to ensure enterprise wide consistency as the final document is created on a centralized server.

## Template-based document processing

Documents can be programmatically built from scratch, combining data from a multitude of data sources with formatting information from templates. Such templates consist of a number of variables, text fields and page layout information. Typically, they are stored in a centralized location, so as to ensure the enterprise-wide constancy of documents, underlining the corporate identity of the enterprise. During the document compilation process, text fields in the template are populated with information from the application's data store.

Today's optimized business processes demand that information is immediately accessible to stakeholders inside and outside a company. Typically, this information needs to be available in a multitude of formats and customizable for every reader. Using server-based solutions, print-ready documents can be generated on-the-fly, with information from a set of databases and user data from a web browser.

The resulting document can be displayed in the browser, saved to disk, stored in a database or sent via e-mail. As documents are generated on-the-fly, they must not be physically stored on the server. This reduces redundancy and, at the same time, gives users direct access to up-to-date, live information.

### Sample: Create invoices on demand

In the last years, there is a recognizable trend that more companies offering their customers to download the invoices from the companies web site or a dedicated portal.

This cuts the invoicing costs dramatically and provides many advantages for the addressee. Latest studies show that the costs for an invoice can be reduced from 10 to 60 USD down to 2 to 4 USD. The reasons are the saved material, printing costs and postage. Additionally, the cash-flow can be optimized, because the invoice is faster at customer site.

A phone company stores the communication data of their customers in a database. Based on this information, the invoices can be created on-the-fly as a full sophisticatedly formatted and optional password secured Adobe PDF file. Using a web-interface, the end-user is able to sort the invoices or the user can define which invoices from the last months should be created.

There are, however, cases in which documents do have to be physically stored on a centralized server. In such situations, developers can create applications that search within these documents or build an index based upon them, thus speeding up subsequent queries. End users can access the documents with a Windows or web-based application.

# WebForms, code-behind or client-side scripting?

## Centralizing the document creation processes

Generally speaking, the complete document creation process is ported to the server. This provides the developer more control and flexibility over the documents. Documents can be sent directly via e-mail from the server or can be printed, put in envelopes and shipped in an outsourced batch process. Server side word processing enables developers to better integrate document creation processes.

This architecture does not inevitably imply that the complete word processing and the editing of documents is limited to the server. It allows developers to finalize the documents on the server or to create the documents on demand server side. Users still should be able to modify documents client side, using a Windows application or a web-based application in a browser.

Planning a new server - or web-based business application can be a difficult task. Many concepts, buzz-words and technologies are currently being propagated by the media: AJAX, WebForms, ASP.NET, code-behind and client side scripting are typical examples.

This paper explains which technology is suitable for which application. Assuming the document creation process has been delegated to the server, the user should be able to modify the template and the document itself.

## ASP.NET: WebForms and code-behind

WebForms are currently introducing a programming model built around server side controls. Web controls render their own user interfaces by generating HTML that is returned to the client and by firing events that are handled by server side scripts. This action takes place on the server, so that virtually any browser can run a WebForms application. The programming involved is similar to WinForms

programming, in which components, combo boxes or buttons can be easily dragged onto a form.

The programming of these web controls is done using Microsoft's .NET programming languages, such as VB.NET or C#. As mentioned above, the programming takes place on the server and can be placed in the ASPX page itself or in a separate file (,code-behind').

TX Text Control .NET Server and TX Text Control ActiveX Server are pure server side components that can be programmed using VB.NET or C# in an ASP.NET web project. These components are not visible - they do not render their own user interface. The TX Text Control server components have been created to modify, create or convert documents without any user-interaction. Templates can be loaded and merged with data from different data sources. They support the following document formats: RTF, MS Word, XML, HTML, TXT and PDF.

## Editing documents in the browser

There are several approaches that developers may take when displaying a document in the end user's web browser. Each approach has certain limitations and thus is not suitable for all applications.

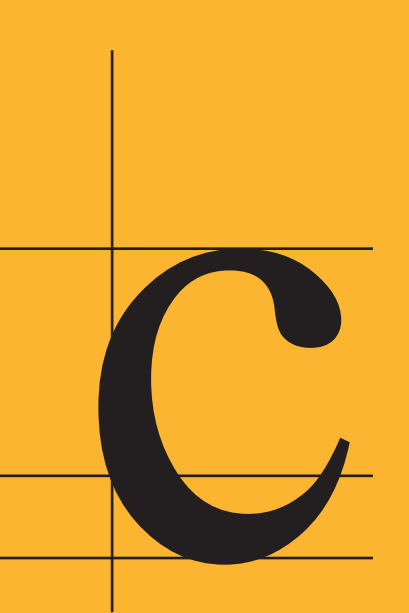
The following table illustrates which technology can be used in conjunction with which TX Text Control server version in a web browser.

If the document is to be edited by the end user, a client side component is required. The above mentioned ASP.NET web controls generate a user interface, using a combination of HTML and a client side scripting language. This allows sophisticated interfaces to be created that can partially simulate Windows applications.



### What is ASP.NET?

ASP.NET is a set of web development technologies. It is dedicated to build dynamic web applications or XML web services. ASP.NET is part of Microsoft's .NET Framework and is the successor to Microsoft's ASP technology.



New technologies, such as AJAX (asynchronous JavaScript and XML) allow more responsive interfaces. AJAX is a web development technology for creating interactive web applications by transferring data to and from the server without reloading the complete page. Using this technique, grid or chart components can be conveniently developed, as the formatting capabilities of HTML and JavaScript are sufficient to show charts or a grid table. For true, WYSIWYG display and editing of complex word processing documents, the AJAX approach is insufficient. Such documents require more display capabilities that browsers currently offer.

Intention				
	Edit full document (WYSIWYG)		Edit text blocks (WYSIWYG)	
	MSIE	Mozilla	MSIE	Mozilla
ActiveX	•		•	•
HTML interface			•	•

### Full WYSIWYG: Client-side controls

The only way to enable the truly correct rendering of complex word processing documents in a browser is by deploying client side controls, such as TX Text Control ActiveX. Such controls are downloaded once from the server to the client through a code signed CAB file. They are then automatically installed into a special folder on the client machine and can be used locally in the browser.

A common question is, whether TX Text Control ActiveX can be used in an ASP.NET project with Visual Studio .NET. Although it can, the control must be programmed using client side scripting and not using C# or VB.NET on the server

If the client side ActiveX is used in Internet Explorer, developers can benefit from the full range of word processing features of TX Text Control without any limitations: The complete word processing document can be modified including headers, footers, text frames or images.

### Editing text blocks

Experience shows that there are many applications in which no complete view of a document is required in the browser. In such cases, only text blocks are edited and not the entirety of the document.

Consider the following typical application:

An application should be developed that produces a print ready invoice. The end user should be able to select an addressee from a data grid and product codes, product descriptions and prices using a combo box or comparable widget from the comfort of his/her browser. The end user should then be able to compose a cover letter and format it at will. Only the inserted text blocks should be editable - not the entire document.

To fulfill these requirements, a simple web control with an HTML interface can be deployed. Using basic browser technologies and HTML formatting, these text blocks can be styled and then merged back into the main document on the server. By generating the final document server side, the enterprise wide consistency of documents can be guaranteed.

The TX Text Control product family offers both a client side control and a server side web control for TX Text Control .NET Server. The web control is not part of TX Text Control .NET Server itself and can be replaced by any other HTML edit web control.

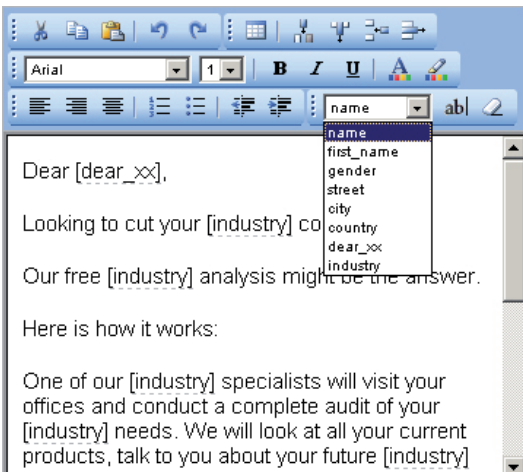
Using TX Web Control, developers can create applications that load text blocks in any supported file format (RTF, MS Word, TXT, TX Text Control format or HTML). Additionally, end users can insert placeholders into their text blocks that are later merged with data from databases or XML files server side.

The following screenshots show the HTML browser interface and the client-side ActiveX:

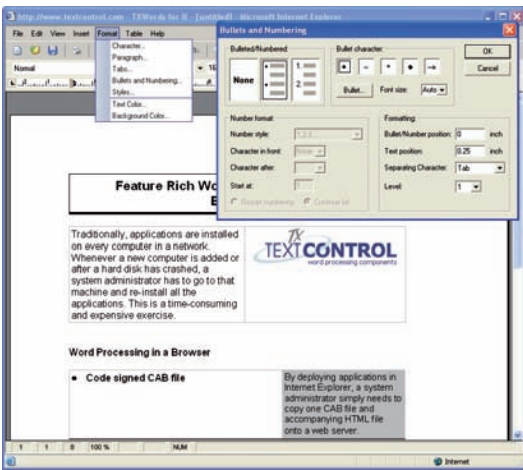
#### Where do I find the TX Web Control?

TX Web Control is not shipped with TX Text Control, but a sample implementation of an HTML edit web control to show the possibilities with TX Text Control .NET Server. Contact the Support Department to get a version of the TX Web Control:

[support@textcontrol.com](mailto:support@textcontrol.com)



**Screenshot:** HTML interface web control



**Screenshot:** Page-oriented client-side ActiveX control

## Conclusion

TX Text Control offers a broad palette of word processing components for different applications and software architectures. By porting critical document creation processes to the server, developers are able to ensure their consistent execution enterprise wide.

Server side word processing applications enables end users to create business documents that are stored centrally on a server. The documents can be further processed by programs such as document management tools or knowledge management software.

TX Text Control .NET Server can be used to create, modify or convert these documents server side within an ASP.NET project, web service, Windows service or a Windows application. Moreover, TX Text Control .NET

Server can merge templates with data from various data sources to create print ready PDF documents. A template designer to create ready-to-use templates, including database text fields ships with TX Text Control .NET Server.

If documents are to be modified in a browser, TX Text Control offers two different solutions: The client side TX Text Control ActiveX for a full WYSIWYG view or an HTML interface web control in conjunction with TX Text Control .NET Server.

## Feasibility Discussion

If you are planning a server-based word processing module in your application, please feel free to contact our engineers to discuss the possibilities. This service is dedicated to developers planning a new project in which word processing is required.

The service is designed to get you the information you need to make a good and speedy decision with little effort on your behalf.

Commonly posed question include:

1. Should my application be web-based, server-based or Windows Forms-based?
2. Which TX Text Control version should I evaluate?
3. How can TX Text Control help me to improve my business processes?

The Feasibility Discussion is part of the TX Text Control support services and is completely free. Feel free to speak to our engineers before reading the documentation, trying different versions of TX Text Control or thinking about the architecture of your software.

Whether you would like to know, which version of the TX Text Control product family fits your current project or whether you need help on deciding upon which development environment can be used with TX Text Control, please talk to one of our engineers.

**support@textcontrol.com**



## The Imaging Source Europe GmbH

Sommerstr. 36  
D-28215 Bremen  
www.textcontrol.com  
Tel. +49 421 3359 10