/\* This JScript function collects the form data and calls the web service. \*/

function generatePDF() {

var httpURL = "", httpURLVirtual = "", rawTxt = "";

/\* Grab form values \*/

httpURL = $get("pdfConvertUrl").value;

httpURLVirtual = $get("pdfConvertUrlVirtual").value;

rawTxt = $get("pdfConvertTxt").value;

/\* Display message until web service responds \*/

$get("pdfMsg").value = "-- Processing your request...";

/\* Call the web service using .Net AJAX \*/

/\*

What these arguments mean:

1. The first three are defined in the web service itself (httpURL, httpURLVirtual, rawTxt).

2. The fourth one indicates the JScript function to call when the web service is successful.

3. The fifth one indicates the JScript function to call when the web service fails.

4. The sixth one indicates data formatting.

Note: Even if your web service does not accept arguments, you still call that web service with

at LEAST the JScript function to call when the web service is successful and the JScript function

to call when the web service fails.

The following .Net AJAX function call, passes the following to the web service:

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<ajaxPDFGenerator xmlns="pdfHTMLDOC">

<httpURL>string</httpURL>

<httpURLVirtual>string</httpURLVirtual>

<rawTxt>string</rawTxt>

</ajaxPDFGenerator>

</soap:Body>

</soap:Envelope>

\*/

PDFAjaxWebService.ajaxPDFGenerator(httpURL, httpURLVirtual, rawTxt, responsePDF, responsePDF, "text/xml");

}

/\*

This JScript function is called after the web service completes. xmlData contains the response data.

Remember, with this implimentation of .Net AJAX, you don't need to manually handle the XML node

crunching (XMLHttpRequest() or ActiveXObject("Msxml2.XMLHTTP"), readystate and so on available under

the Javascript DOM) commonly employed in non .Net environments.

The following function, receives the following from the web service (which .Net AJAX automatically

extracts the data from):

<ajaxPDFGeneratorResponse xmlns="pdfHTMLDOC">

<ajaxPDFGeneratorResult>

<generationResultsBin>

<string>string</string>

<string>string</string>

</generationResultsBin>

<generationResultsMsg>string</generationResultsMsg>

<generationResultsURL>string</generationResultsURL>

</ajaxPDFGeneratorResult>

</ajaxPDFGeneratorResponse>

\*/

function responsePDF(xmlData) {

var compiledMsg = "";

compiledMsg = "<generationResultsMsg>" + xmlData.generationResultsMsg + "</generationResultsMsg>\n";

compiledMsg = compiledMsg + "<generationResultsURL>" + xmlData.generationResultsURL + "</generationResultsURL>\n";

if (xmlData.generationResultsBin.length > 0) {

compiledMsg = compiledMsg + "<generationResultsBin>One or more alerts should have shown the values of the child nodes of this node. Depending on what data is being rendered, you may or may not see the full contents of a child node.</generationResultsBin>\n";

$get("pdfMsg").value = compiledMsg;

for (x = 0; x < xmlData.generationResultsBin.length; x++) { alert("#" + x + " Child Node Value:\n" + xmlData.generationResultsBin[x]); }

}

else { $get("pdfMsg").value = compiledMsg; }

}

/\* Populate the textbox with HTML code \*/

window.onload = function() {

$get("pdfConvertTxt").value = "<html>\n<head>\n<title>Test</title>\n</head>\n<body>\nThis is a test.</body>\n</html>\n";

}