



Welcome to the DBTS Form Processor Extension for WYSIWYG WB!

DB Technosystems is well known for quality, custom made software. The main fields of application are database driven scripts like membership scripts, Shopping Carts, directory scripts and other similar.

The WB Form Processor extension is the result of a continuously rising request for a simple, trouble free, multi featured form processing script. It is essential, but flexible and powerful. The output is highly customizable and is continuously enriched with additional features.

This extensions is part of a suite of 3, together with the DBTS Preview and the DBTS reCAPTCHA extensions.

The DBTS Preview is a completely customizable extension that works with the DBTS Form Processor Extension. It will allow you to create Preview Pages and Data Merge Confirmation Pages simply dragging the Preview Object in your pages.

The DBTS reCAPTCHA is a very simple extension that allows you to embed the famous reCAPTCHA in your forms with just a few mouse clicks.

In the next pages you will find the complete installation and usage instructions. Please read them carefully and you will be able to install and use it in just a few minutes.

Thank you

DBTechnosystems.com

Update by Pablo

Because this extension is no longer maintained by George, but is still useful to many users, I have updated the internal code to make it compatible with PHP 8.

Requirements

DBTS Form Processor requires PHP 7.4 or higher support.

Safe Mode must be OFF and the folder in which they are used should have read and write permissions if you want to take advantage of the advanced features like file storage and file uploads.

The extension will work with WYSIWYG Web Builder 20 and higher versions.

Please note that support for this extension is discontinued and no further development will take place.

Installation

The DBTS Form processor is installed just as any other WB extension: Through the extension Manager, you can perform either an Automatic Installation or a manual installation after you have downloaded the zip with the extension files.

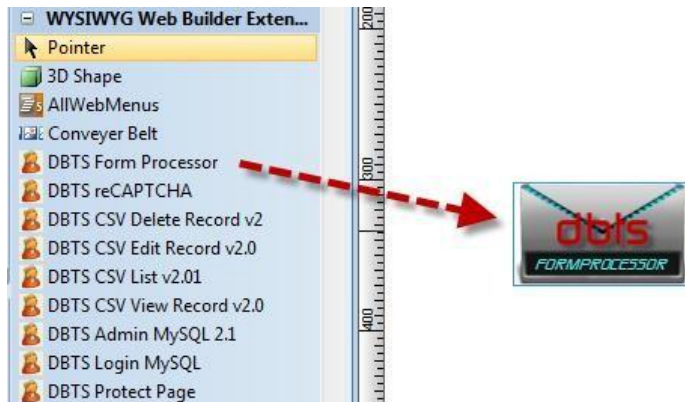
Features

The DBTS Form processor extension allows you to process forms created with WB.

However, if you create a blank page WB and add the DBTS Form Processor object in it, you can use it to process any other submission from any of your pages or even from another website. It will add the necessary PHP code in the Start of Page of your page code. This code will process the form submission, if a submission has taken place, or will remain inert if no submission has taken place.

Using the DBTS Form Processor Extension

The DBTS Form processor extension will add an icon in your extensions Toolbar.



To use the extension simply click and place the Form Processor object in your page. The position can be anywhere, as there is no visual output in the published page.

To setup the processor properties double click the Form processor Object so the properties dialog box will appear:

A. Administrator and General details	
Form ID	1
From Email Address	mail@dbtechnosystems.com
Admin Name To	George
Administrator Email	info@dbtechnosystems.com
CC email address	
BCC email address	
Visitors Email field name	email
Character Set	UTF8
Success URL	success.php
Error URL	error.php
Your Timezone (see Manual)	Europe/Rome

B. Actions	
Send Administrator Mail ?	Yes
Send Autoresponder Mail ?	Yes
Store data in CSV File ?	Yes
Store data in Database Table ?	Yes

DBTechnosystems
<http://www.dbtechnosystems.com>

OK Cancel

The settings are organized by logical category.

A. Administrator and General Details:

Form ID: DBTS Form Processor allows you multiple forms on the same page. If you want to have multiple forms, EACH form must include a hidden field, named formid (case sensitive) . Its value MUST be the same as the Form ID value that you will type in this field of the Form Processor. We STRONGLY recommend that you use a form ID string (number or alphanumeric string, no special characters or symbols) or your processor will process any submission.

"From" email address: The email address that will be used as "from" in the mails sent by the processor. It SHOULD be different from the email address where you want to receive the mail (see next point 3.). The WB built in from processor uses the visitor's email address as "from". If a visitor doesn't fill in his email address no "from" email address exists, and in most cases the mail would be blocked by your ISP. To avoid this, in such cases, the built in form processor used your own email address as "from" address. However, spammers usually use your own email address as "from". This has obliged ISPs to add a special filter that will reject emails that have the same "from" and "to" email addresses. This is the reason why we have added this alternative "from" email address. With most hosting companies, you can use an whatever email address, even one that's not within your own domain, but, due to issues with some hosting companies (Yahoo, GoDaddy etc) it is necessary that this is a real email address within your own domain. You can have as "from" something like contactform@yourdomain.com . You need not worry about the "Reply To" address, if the user has filled in a valid email address, it will be correctly used as such.

Administrator Email: Here you can set the email address where you will receive the form data. You can feed in multiple email addresses, separated by a comma.

CC Email: you can add emails addresses where a CC email will be sent. If you need multiple CC emails, use a comma to separate them.

BCC Email: you can add email addresses where a BCC email will be sent. If you need multiple BCC emails, use a comma to separate them.

Visitor's email field name: many users need to name the "email" field in some other way (typical example: a non English language). This was not possible till now, or the autoresponder would not work. You can now name this field whatever you like, as long as you fill in the related property in the DBTS Form Processor properties.

Character set: you can set particular character sets to be used for the mail. By default UTF8 will be used.

Allow URLs: Setting it to yes will parse the URLs in active links, in the automated HTML email. Please note that this will only happen if your visitor types the http:// .

Success URL and error URL: these are the urls of the pages where the script will redirect in case of successful processing or in case of error. Please note that the script will only redirect to the success page if you have set it to redirect. This is because for some applications, you might simply want the script to exit. Regarding the Error page, please note that it can be a simple .html (or .php) page, with your standard page elements (header, footer, logo, images etc) but, in order to have a correct error reporting, the error page should have the following text in it: ##error##

This text will be replaced by the actual error message. Make the text box large enough.

10.Timezone: you can set your timezone so the times / dates captured by the script will be correct for your country / timezone and not the default (server) ones. For a list of valid timezone settings, please visit <http://php.net/manual/en/timezones.php>

However, Keith (alias Star57 on the WB forum), has created a similar facility for public use. You can find it at <http://www.star57.com/dbfp/index.php> . Quite easy and intuitive to use.

11.Time formatting: you can format the date and time that is included in the mails or stored in the database. For advanced users only. Please see <http://www.php.net/manual/en/function.date.php> for details. If you type ANYTHING here, it will override the default Time of Submission format, that is like: Wednesday 3rd August 2011, 2:56 AM.

This must be a valid date format, or you may have unexpected results.

B. Actions:

This section allows you to set what you want the script to do.

Send Administrator Email: you can select whether or not you want the script to send you an email.

Send Autoresponder Email: you can select whether or not you want the script to send an automated autoresponder email to the visitor that has submitted your form. This email will be suppressed, even if set to yes, if no visitor email is found.

Store Data in CSV File: you can select if you wish the data to be stored in a CSV file on the server. Please note that the script will add three specific fields to the records stored in the file: Submission Date, IP Address, and Timestamp. Also please note that the files created include, for your convenience, the field names. If, in the future, you add further fields, the script will add the new field names automatically.

Store Data in the Database: This new feature will allow you to store data directly from your form into a database table. You will need to create the database and the user / password, and grant this user the necessary permissions. You don't need to create the table, the script will create it for you. It will also add more columns in the table, if in the future you add fields in the form.

C. Further Posting:

This new feature allows you to "POST" the form data (that is, send the same data using the POST method) to any other page or script you wish to. It does it in "silent" mode, that is, it will not open the script in a new or in the same window.

Post details to Another Script? : you can set the processor to send the same details submitted by the form, to yet another script. In example, you can forward details to PayPal. Select "Yes" if you wish to do so.

URL to Post to: this is the full URL in domain.com/scriptname.php format where you want to post the details, WITHOUT http://www.

Please note that at the moment, only data is forwarded, whilst uploaded files are not.

D. Behavior:

Copy Field Values in Session / Copy Session values in Fields (there on referred to -for briefness- as Copy in Session / Copy from Session). This feature will allow you to copy the POST keys and values into session variables. This allows for many tasks to be performed. The simplest one is Multipage Forms! You can create your forms normally and simply process them with our processor. If this is the first form of the Multipage form, you should set the script to do NOTHING except than "Copy in Session". If it is an intermediate form, you should disable all other features and only enable BOTH "Copy in Session" and "Copy from Session". Last, if this is the last page of the Multipage form, you should perform all other necessary tasks (mails, storing etc) and enable the "Copy from Session".

Clear Session Values before Ending (there on referred to -for briefness- as Clear Session): If you are using the two previous features, it is wise that you clear the session values just at the last page of your multipage form or your Preview Page.

Generally speaking, if your data MUST remain available in the memory for further processing, this must be set to "No". If, instead, the data must be cleared from the memory after processing is done, the "Clear Session" value must be "Yes".

After Processing Behavior: you can select if you wish to Display Page Without Refreshing, Redirect to another page, Include another script or page, or simply Exit the script.

'Display the page without refresh' allows you to display the same page whilst it preserves all the posted variables. This way you can display info on the page, if you like. Property for Advanced users only!

'Exit' will cause the script to exit after processing. This is also an option for advanced users who need to perform specific actions and know how to. Property for Advanced users only!

'Include' allows you to include further code for post processing. It needs to be valid PHP file, and is also intended for advanced users. If you select this option, the script, after processing, will 'include' the file that is specified in the next field (see point nr. 15). Property for Advanced users only! 'Redirect' is the most common selection, to redirect a user to the 'success' page.

Page or script to include before Script end: This is the script that will be included just before the script end if in the After Processing Behavior you have selected "Include". There is an "exit" command AFTER the include. Using this include will NOT allow you to redirect or clear the session variables after the 'include' execution. If you have such requirements, you should use the 'End of Script include' in the 'Includes' section. Property for Advanced users only!

Include Null Fields: Here you can select if you want the script to include the null fields (fields where the user didn't fill in anything) or not.

If you select to include the null fields, here you can set the text that will appear as field value. It can be some text like "No user input" or it can simply be blank.

Excluded fields: you can exclude some fields from the mails, if you so wish. Type the field names, separated by a comma.

Allow URLs: this feature will convert any valid URL in active hyperlinks. This will only happen if the user has typed the full (with http://) URL.

Include Info in the Autoresponder email: you can select if the autoresponder email should include a copy of the info submitted. Please note that for uploaded files the script will display only the file name without revealing the path where the files have been stored.

10.Store Uploaded files on the server: Select if you want the uploaded files to be stored on the server. If you allow uploads, we STRONGLY recommend that you store them on the server, just in case your ISP blocks the mail with the attachments due to oversize.

11.Attachments: you can select to receive the uploaded files as attachments to your email.

12. Maximum Total Attachments Size: If you know what's the limit for email size that your ISP allows, you can set this limit here. If the Total Attachments Size is higher, the script will not attach the files to avoid your email been blocked.

E. Captcha:

WWB Captcha Rewrite: This feature ONLY regards the standard WB captcha. DBTS Form Processor will work with the standard WB captcha as it uses its own verification mechanism before the processor even comes into scene. However, in this case, the Wrong Code message would appear in a rather ugly white page, with a "Back" link. Further more, if we left it like this, we would not be able to use the method described in the "Tips & Tricks" section to make our form preserve the values that have been filled in, in case of wrong captcha. This is why we have added this "Captcha Rewrite" feature. Setting this to "Yes" will result in rewriting the entire page on the server without the default verification code. This way the DBTS Form Processor built in captcha verification will become active. It will also abort the default white error message page that is displayed, and will use instead your own error page to report the error in the captcha. Also note that this

replacement will ONLY take place in occasion of the first successful submission through the form, so it is wise that you test the form yourself, so this replacement takes place. If, at some point, you republish the page, please repeat the procedure.

Don't set this to "Yes" if you are using reCAPTCHA.

Some servers use non-standard eol (end of line) characters. This happens usually on Windows servers. In such case, the page re-write will fail and will cause (usually) either a blank page to display or an error message. If you see any of the two, please disable the "Re-write Captcha" feature and publish the page again.

A good alternative is to use real time validation for the WWB captcha, in which case you don't need to enable the Captcha Rewrite feature.

WWB Captcha Error message: The error message that will appear if wrong Captcha value is entered.

Force reCAPTCHA: Setting this to "Yes" will force the DBTS Form Processor to check and verify the reCAPTCHA field value. reCAPTCHA is considered one of the best captcha webservices, offering a high level of security. You may use either the WWB built in reCAPTCHA or the DBTS reCAPTCHA extension. The later has the advantage that it allows language selection.

To use reCAPTCHA you need to first create a reCAPTCHA account, and generate your Private and Public Keys. To do so please visit

<http://recaptcha.net/>

reCAPTCHA Errors: The reCAPTCHA value is verified using reCAPTCHA's API and specific servers used for this purpose. However, when an error occurs during procedure, the reCAPTCHA server returns back standard strings in

English, which are not very human friendly. Therefore we have implemented an error handler that will report the error in plain human friendly language. You can customize these errors as you prefer and, of course, provide the error message in whichever language you like.

reCAPTCHA failure behavior: in very rare occasions, either because there are server issues or connection errors, the reCAPTCHA verification might fail. With this property you may select if you want the script to allow the submission (which would otherwise fail, so you would lose the data submitted by your visitor) or allow the submission anyway.

Reverse Captcha: The idea behind this is to let spambots "identify" themselves as such instead of requiring users to "identify themselves as humans. Actually we suggest that you use both methods combined.

The verification consists in the following:

Spambots "read" your form, and then they fill in all fields with random values. Usually textarea fields are filled in with spam content whilst textbox fields are filled in with random values. To avoid spam submissions, you add, in your forms, two invisible (NOT hidden) fields. For the purposes of this manual, we will suppose that you call them "ver1" and "ver2" (but you can use any field names you like). You then assign to the first one an whatever initial value (in example, "this is my verification string"), whilst the second one must be left blank. A human will never see these fields, so he will never fill them with any values, thus preserving the initial values assigned by you. A spambot instead that "reads" your page, will fill these fields with random spam values. So if the script "sees" that these fields are filled in with values different than the initial ones you have set, it will understand that this is a spambot submission and will behave accordingly.

You should fill in all the properties of this section if you want the script to behave correctly.

Use reverse captcha ? This property allows you to set the processor check for the existence and compliance of a Reverse Captcha.

Non empty field name: this is the name of the field that has a preset value. Avoid names that are composed by more than one words. If necessary, use underscores.

Empty field name: this is the name of the field that should remain blank. Same considerations about naming.

Reverse captcha behavior Here you can select if you wish to display an error message, which actually is useless since we are dealing with a spambot, or simply redirect to the “success page” without actually submitting the data (recommended).

Reverse captcha error message: If you have set, in the previous property, the Behavior to display the Error Page, please type the Error message that will be displayed.

F. Departments:

You can set the processor to send the mail to different email addresses, depending on the user selection. In example, you may have a contact form, and you might want to allow the user to send the mail to your Sales, or your Support, or your Billing department. What you need to do is to add a dropdown (or a set of Radio buttons) that the user can select from. Set correctly the properties of this section:

Send email to specific department ? Select Yes to enable this property

Department field : the name of the form field used for the user selection.

Departments Settings : here you can type the name as well as the email address of each recipient. The name will be used as “To Name” for the email. The email address will be used as “address to”, in other words the mail will be sent to this email address. You can have multiple email addresses, separating them by commas.

If you have enabled this property but the user doesn’t select any department, the mail will be sent to the Administrator email address (see above, point A.3.) G. Banned IP addresses:

You can ban certain IP addresses if you receive spam mails. You have to set the following properties:

Behavior: Here you can select if you wish to display an error message, to let the submitter know he is banned, OR you can redirect him to the success page, to let him think that his submission has been successful. This last option is recommended as, if the spammer sees that he is banned, might change his IP address (most spammers can do this).

Error Message: If you have set, in the previous property, the Behavior to display the Error Page, please type the Error message that will be displayed.

Banned IP addresses: Please set the IP addresses you need to ban. You can type a short name or description for each one. This is only used as a reminder for you. You can also ban entire IP ranges, by simply typing the first part of the IP range.

H. Logo Settings:

You can add your own logo image on top of the emails created by the script. The necessary settings are:

Image URL: your image should be hosted in the same or a subfolder of the folder where your form is, and you need to provide the RELATIVE path. In example, if the image is in the same folder as the form, the value will be simply “logo.jpg”. If it is in a subfolder called “images”, you must type in “images/logo.jpg”. Alternative text : this is just the same as the alternative text for images in your website pages.

Link: you can link the logo image to your site or whatever other site you like.

Logo Area Background Color: you can set the background color.

Logo image Horizontal Alignment: can be set to left, center, right.

Please note that the image will be embedded in the mail itself, so neither you or your visitor will not have annoying warnings from your email clients. I. Visual Settings:

The extension creates a table with the submitted info as well as the headers and footers of your choice.

The mail table that is created has 5 areas:

The Logo area (optional)

The header area (optional)

The info area (optional in the Autoresponder)

The footer area (optional)

The technical info area (optional)

Please refer to the following image to see what each area contains:

The image shows a screenshot of an email form with several labeled areas:

- Logo Area:** Points to the "db Technosystems" logo at the top.
- Header Area:** Points to the text "There has been a new submission from our Extension Demo Form. Here is the info submitted:".
- Form Data Area:** Points to the table containing submission details.
- Footer Area:** Points to the text "Please review and act accordingly.".
- Visitor Info Area:** Points to the technical information section.

There has been a new submission from our Extension Demo Form. Here is the info submitted:	
First Name	George
Last Name	Naval
Address	My Address
City	Los Angeles
State	FL
Country	United States
Site	http://www.dbtechnosystems.com
Email	info@navaldesign.it
Telephone	33333333
Fax	44444444
Mobile	55555555
Comments	Testing the new Form Processor Extension
Priority	Normal
Answer By	Land line
How Did You Find Us	WYSIWYG WB Forum
Please review and act accordingly.	
Time of submission :	Wednesday 15th July 2009, 3:34 PM
Browser :	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; GTB6; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; SLCC1; .NET CLR 2.0.50727; OfficeLiveConnector.1.3; OfficeLivePatch.0.0; .NET CLR 3.5.30729; .NET CLR 3.0.30618)
IP Address :	79.55.160.173

For each of these areas, you can set the Text color and the background color.

For the info area and the technical info area, you can set separately the color and the background color,

The last section, Visitor Info Area, has the same settings of the Form Data Area.

You can also set the page background color, though this will only be visible if you use a desktop email client, as webmail software doesn't yet support, in most cases, css.

You can set the width, cellspacing, cellpadding, border width and color of the table that is created.

– Labels:

You can customize the three labels used for the technical info section of the script: Time of Submission, IP Address, and Browser.

- Mail settings and Autoresponder settings:

As we already explained, the mails created have a header, a form data and a footer section.

The form data section is created automatically by the form processor and it includes the form submitted info.

The Header and the Footer text is customizable for both mails, and this is where you can set this text.

For the autoresponder “From” you can use any email address you like, or even a `no_reply@yourdomain.com` address. It’s up to you. However, some hosting companies (GoDaddy and Yahoo, in example) have limitations regarding email addresses used as “from” so if you test and no autoresponder email is sent, we recommend to use a real, existing email address, within your own domain.

You can disable the inclusion of the visitor IP address, browser and Time of Submission info, disabling these features in the email settings.

New feature: You can now use your own HTML (or PHP) template for the Administrator and autoresponder emails. The script will “read” the template and use it for the mails. This feature is quite useful if you need to send an order or shopping cart confirmation mail. It is important that your template is a valid PHP / HTML email. Furthermore, whilst modern email clients like Windows live mail support CSS and will display the layout exactly as you see it in your browser, online webmail interfaces usually don’t (Gmail, Yahoo mail etc). Therefore it is important that your template is created with tables.

New feature: you can now use one or more of the data submitted through the form as “From Name in the Administrator email. This way you may have the visitor’s first and last name appear as “From Name” in your mail. Just fill in the related property the field names separated by comma (or commas)

L. File Uploads:

Here you can set the allowed file extensions, the maximum filesize, the folder (it will be autocreated) where the files will be uploaded, and (we strongly recommend) whether to add a security prefix to the file names. A security prefix will protect you from hackers that will try to upload a file on your site and then try to run it. Please note that the WB validation for file extensions is Javascript, so it will not protect you from someone that has Javascript disabled in his browser. Also name your “uploads” folder something strange.

In the near future we plan to add PHP validation to the script for most possible validation types.

The script will include the path (download URL) of the uploaded files in the Administrator’s email.

Please note that the script will ONLY accept files whose extensions are included in the list of allowed extensions. File extensions are case insensitive, so there is no issue with files having extensions in lower or uppercase characters.

Important Security Notice: PLEASE do NOT open files on your browser if the file extension seems strange or you don’t know it. Do NOT open in your browser uploaded file links if the extension is something like asp, php, js, exe, or ANYTHING else that you don’t know if it is secure. It will be better to download them with FTP and examine them on your own computer.

M. CSV File Form Data Storage:

You can set the script to store the form submissions in a file on your hosting account.

If you select to do so, you can also select the type of separator you want to be used, among the following two types:

Semicolon “;” Recommended.

Comma “,”

If you need to preserve formatting of fields like textareas and avoid line breaks caused by data that contains the delimiter in the text typed in your form, I suggest you set the “Use double quotes in field values” property to “Yes”.

Please note that the script will add Date of Submission, IP Address and Timestamp at the beginning of each record. It will also add the Columns (fields) names. In case new fields are added in the future, it will update the column names adding the new columns as required.

Uploaded files are displayed as URLs in the csv file. N. Database Storing:

You can set the script to store the info submitted in a database table. In this section, you can provide the necessary database details: Database name, Table name, username and password, as well as Database Host. Please consult your hosting company documentation on creating a MySQL Database.

The script will automatically create the table if it doesn't exist, and will automatically add columns if necessary. Please do NOT use whitespaces or symbols for the table name. O. Mail Engine Settings:

The script uses the famous PHPMailer Class as mailing engine. This class allows mailing using the following modes:

Mail() function

SMTP with or without authentication

Sendmail and Qmail

Using authenticated SMTP you can even use your Gmail account if you wish. However I don't see why, if your hosting account allows you to send SMTP mails.

Authenticated SMTP means that the mailserver will verify the SMTP “from” email address and password. This email address MUST exist, so you SHOULD create an email address, if you haven't yet setup one, and use that email address and its password as SMTP “from” email address and “password”.

By default the script will use the mail() function.

SSL and TLS: you can now use SSL (Secure Socket Layer) and TLS (Transport Layer Security) for the mailserver if required. Just check the relevant option.

The script allows, as said, to send mails through SMTP.

With most hosting companies, this is the ideal solution (although, most of them will now also allow to send mail through the PHP mail() function.

Some companies will require that you use SMTP authentication, to send email through SMTP. This is quite possible with the DBTS Form Processor, as long as you use the correct details.

The SMTP server usually is something like ‘localhost’ or ‘smtp.domainname.com’.

The first is (usually) valid if you are using your own domain smtp server, the second one is valid if you are using an external smtp server like Gmail.

The smtp username is your own email address on the server you are calling. In example, if using your own domain smtp server, it will be you@yourdomain.com , where this is an existing email account. The smtp password, will be the password you have setup for this email account.

Please find below a list of popular providers that you can send email through SMTP, using the settings mentioned below: (source: PHPMailer site, link:

http://phpmailer.worxware.com/index.php?pg=tip_srvrs

P. Custom Processing and Includes (Advanced Users Only!):

The Extension allows you to add custom PHP code in strategic points of the original script code. These are:

Start of script

Before Form Data reception

After Form Data reception

Before Error reporting

Before emailing the Administrator

Before emailing the user (if autoresponder is enabled)

End of script, before redirection to the success page.

This way you can perform lots of actions using your own code and including it in the most appropriate position inside the original code. Please take a look at the “Tips & Tricks” section to see classic use of includes.

The Form and its settings:

DBTS Form Processor does NOT create the form for you!

You need to create a form in your page using the WB Form Wizard or the WB form tools. Once you have created the form, make sure to respect the following settings:

The form page MUST be published as PHP.

The form action MUST be set to be “<?php echo basename(__FILE__);?>

The form encoding type MUST be “multipart/form-data”

The submit button MUST be named “submit” though its value can be anything (like “Send”, “Send Info Now” or anything else).

Reserved field names:

You can NOT use the following field names as they are reserved for the script internal scopes- Most common are: formid , count , captcha_code, Submission Date, IP Address, Timestamp, File, Autoresponder, bodystart, bodyend, subject, username (dedicated for use with the WWB login script).

Please look at Annex A for a detailed list of script internal variable names.

Tips & Tricks:

Setting up correctly your forms field names:

Though you may name your fields (almost) whatever you like, please take this piece of advice: ONLY use field names that don't contain whitespaces or symbols. Don't use quotes ' or double quotes ". Don't use whitespaces, use underscores. They will be stripped of by the script before output (and will avoid you LOTS of troubles). Don't use special characters and symbols like "@", "\$", "<", ">", "-", "\", and dots. Some of the advanced features of this processor make use of specific PHP functions. The above characters / symbols have a particular meaning for PHP and might cause unexpected behavior.

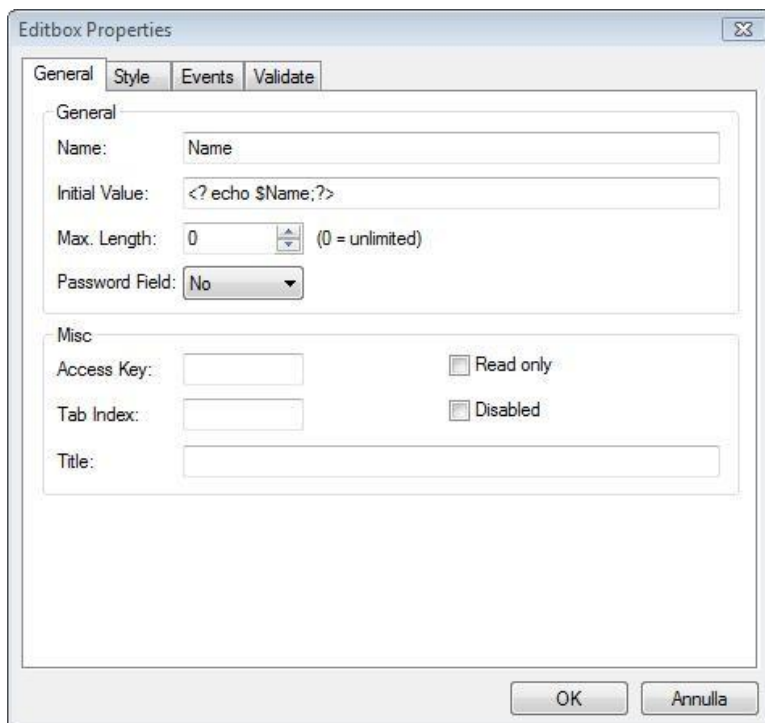
Preserving your form values after submission with wrong captcha value:

Well, this has been a general request by WB users. Having a captcha in your page unfortunately affects the "Back" feature of browsers, meaning that they don't preserve the filled in values. Unfortunately this is not possible with the captcha verification code that WB adds to the page, because WB places the relevant code before anything else in the page, so it doesn't leave any margin for other code to take control. So you need to set, in the DBTS Form Processor, the "Rewrite Captcha" to "Yes". This will remove from your page the code automatically added by WB and will activate the internal DBTS Form Processor verification code, which is practically the same, but uses your Error Page to report the wrong code error.

The SAME STEPS must be followed also in case you use the DBTS reCAPTCHA extension, with the only difference that you must set the "WWB Captcha Rewrite" to "No".

To make your form preserve the data submitted when returning to the form after a wrong captcha submission (or any other error) you need to act on your form fields as follows:

Text Fields and Text Areas: Just fill in the "Initial Value" field, the code: `<?php echo $Name;?>` where Name is the field name



The screenshot shows the 'Editbox Properties' dialog box with the 'General' tab selected. The 'Name' field is set to 'Name'. The 'Initial Value' field contains the PHP code '<?php echo \$Name;?>'. The 'Max. Length' is set to '0' with a note '(0 = unlimited)'. The 'Password Field' is set to 'No'. In the 'Misc' section, 'Access Key' and 'Tab Index' are empty, and 'Read only' and 'Disabled' are unchecked. The 'Title' field is empty. 'OK' and 'Annulla' buttons are at the bottom.

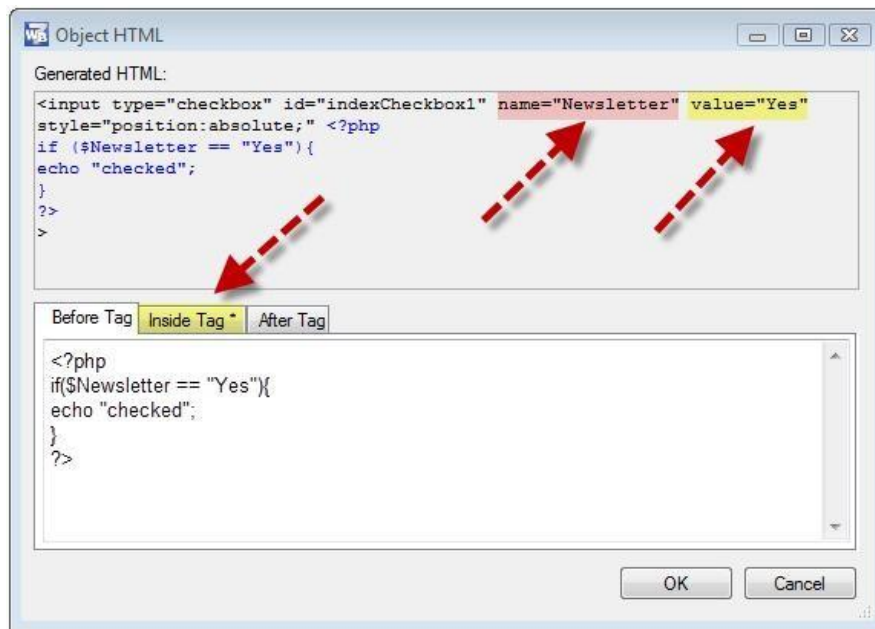
General	Style	Events	Validate
General			
Name:	Name		
Initial Value:	<?php echo \$Name;?>		
Max. Length:	0	(0 = unlimited)	
Password Field:	No		
Misc			
Access Key:		☐ Read only	
Tab Index:		☐ Disabled	
Title:			

OK Annulla

Radio Buttons and checkboxes: Here we have a bit of a more complicated situation. Originally, the radio Button or Checkbox must NOT be selected. If you make a Radio button or a checkbox selected right from the start, you can not make it “remember” the user selected status, as the “selected” status you have originally set will override any choice made by the visitor. With this said, you must right click the radio or check button, and paste the following code in the Inside Tag of it’s HTML:

```
<?php
if ($Newsletter == "Yes"){ echo "checked";
}
?>
```

Newsletter is the name of the radio button or checkbox, and Yes is the value that the specific field will have if selected. Look at the image below that better explains this:



Of course, if you have a group of Radio Buttons or checkboxes, the next (in this example, the “No”) button code should be something like:

```
<?php
if ($Newsletter == "No"){ echo "checked";
}
?>
```

3. Drop Down lists (listboxes - comboboxes): this is also another complicated task. First of all, let me say that there is nothing you can EASILY do with WWB’s listboxes if you have multiple selections enabled, unless you use case-specific code, but this is not what we are doing here. Since listboxes have multiple values to chose from, there is actually no way to make a specific value (or values) selected. So we will do a trick: we will ADD, before all hardcoded values, the one that has been previously selected by the user. To achieve this, right click the listbox and add the following code in the Inside tag of its HTML:

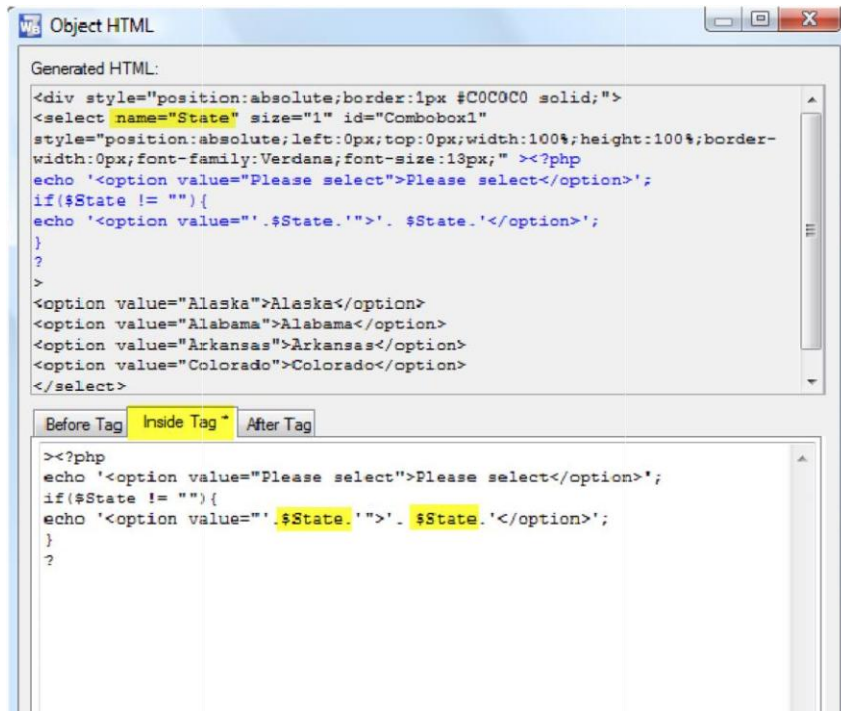
4.

```
><?php
echo '<option value="Please select">Please select</option>'; if($State != ""){
echo '<option value="'. $State. '">'. $State. '</option>'; } ?
```

Of course, you must replace the field name used here, State with the actual name of your field.

Please note that you should NOT set in the WWB listbox the usual “Please select” option. Instead, you can set it directly in the above code, replacing the text “Please select” (in both positions) with whatever you like.

Please refer to the following image so the above result clear:



5. Upload fields: Browsers and Javascript are designed so that upload fields values can not be preset for security reasons.

Custom Processing:

This is a very easy task with the DBTS Form Processor. You can use the “Custom Processing and Includes” section to add your own PHP code in strategic points of the script.

For each of these points of the script you can insert VALID php code. Please note that single quotes in your code MUST be backslashed.

I will give here just a “logical” example, as I believe that this feature requires experience in PHP coding, and, whoever has this type of experience, will easily be able to understand and implement it.

Suppose that you have a “email” field in your form, and you need to add PHP validation to your script (a feature that is anyway planned for the next major version), to verify that this email address doesn’t already exist in the database.

DBTS Form Processor uploads on the server a file with certain functions. Among these, there is a function that performs exactly this: it checks a given column in a table in the database, and checks to see if the value submitted through the form, already exists or not.

This function is `check_unique_in_db($string, $table, $column, $exist_error)`. The arguments passed are the string to search for, the table, the column where it needs to look at, and the error message or value to be returned. To achieve this you need to add the following PHP code:


```
$error .= check_unique_in_db($_POST["email"], "users", "email", "Sorry, this password already exists");
```

The above tells the script to check in the (same) database it uses, in the “users” table, and in the “email” column, to see if the value submitted through the “email” field of the form, exists.

If it does, it will return the error “Sorry, this password already exists”.

Add, in the “Before Error Reporting Include” the above code and the extension will use it to accomplish the desired task.

Of course, if you wish, you can use an “include 'filename.php' ” command to include entire files, if required.

Experienced users will find it easy to analyze the DBTS Form Processor built in functions (download the file from your server!) and see how to use the “includes”.

Creating multipage forms:

This is a very easy task with the DBTS Form Processor.

Create your forms as you would normally do. Please follow the steps described below:

Add the DBTS Form processor object in all forms that contain forms.

Set the form processor NOT to do anything at all, in all pages except the last one.

In the first form, set the “Copy in Session” to “Yes”.

In intermediate forms, set BOTH the “Copy in Session” and the “Copy From Session” to “Yes”

In the last form page, set the “Copy from Session” to “Yes”, and also set the “Clear Session” to “Yes”.

The last page will have in “memory” all the field values posted by all pages, in the correct order. So you can, in this last page, set all the actions that the processor should perform, like emailing, uploading files, storing in file and / or DB, etc.

Limitation: If you want to allow file uploads in a multipage form, the file upload fields MUST be in the last page form.

You should set the “After Processing” behavior of each form, to be “Redirect” and its “Success URL” should be the next form page.

Creating a Success Page that contains the submitted data (data merge):

As we have already seen, if you want your visitor’s submitted info formatted in a table, you can simply use the DBTS Preview Object. Further info is available with the manual of that extension. However, sometimes, you might prefer a free layout that will contain the submitted info.

DBTS Preview makes available the info submitted, just before the script end. If, in example, you have an editbox named “Name”, the php variable `$Name` contains the value submitted by the visitor. It is essential that the success page is published as .php

Let’s suppose that you have named your success page “success”, so you have published it as “success.php”.

To display the value of this (and any other) variable in a WB page, it is enough to do the following:

Form Page:

Set the DBTS Form Processor to do what you want: emails, csv storage, database storage etc. Set the “Copy in Session” to “Yes” and set the “Success Url” to be your success page. Set the “Clear Session” to “No”.

Success Page:

Add the DBTS Preview object in the page. Since you don’t want the standard info display table, but you want a free layout, set the “Display Info Table” to “No”.

Add the text you like to display containing the variable, just as in the following line:

Hello '.\$Name.', thank you for your submission. '

Right click the text, and add the following in the Before Tag:

```
<?php echo' and in the After Tag:
';
?>
```

Please pay attention to the single quotes and the dots.

Add as many variables as you like.

Please note that if you have used fields consisting in more than one words, with whitespaces between them, the variables will have underscores between the words. In example, for a field named “How did you find us” the relevant variable will be “\$How_did_you_find_us”

Save the page and publish it. It is important that it is published as .php.

That’s all. The script will process the info and it will redirect to the success page that you have published, and if it is correctly built, will display the values submitted by the visitor.

Creating a Preview Page before Final Submission

Sometimes it is necessary that the user reviews his input just before he makes the final submission. It is simple enough with DBTS Form Processor and DBTS Preview.

In the form page add the DBTS Form Processor Object and only set “Copy in Session” to “Yes”, without any further processing. Set its “Success URL” to be the Preview page.

In the preview page, add the DBTS Preview object. Double click it to set the visual properties as desired, so it will display the submitted values. Under the DBTS Preview Object add an empty form object. In this form add ONLY a submit button, and a “Back” button so that the user can go back in the first page if the info needs editing. It is not necessary that it actually is a “Back” button, it can be a simple link button or text or image linked to form page. All of the above will work.

Since we want the data to remain available to the Form Processor for the final submission, set the “Clear Session” setting to “No”.

Add the DBTS Form Processor Object and set it to do what you want (mail, autoresponder, etc).

Set the “Copy From Session” to “Yes”.

Set the “Clear Session” to “Yes” so it will clear the session values after this final submission.

That’s all, you are ready!

Support:

You can post questions on the WYSIWYG forum . But please first read this manual carefully, the answer to your question is probably already contained in this manual.

Support that goes BEYOND forum help and requires entering your hosting account to solve any issues, can only be provided on an agreed upon, paid for, basis.

License:

The PHP code used in this extension has been developed by DBTechnosystems.com . The extension is freely available to all WWB users, and can be embedded in any work, personal or commercial. You are not allowed to redistribute or sell this extension. Please address to our download page anyone interested in using this extension:

http://www.dbtechnosystems.com/wb6/forms/dbts_form_processor.php

This software is delivered AS IS. By installing and using it, you agree that Pablo Software Solutions or DB Technosystems will not be held liable for ANY damage, of ANY type, caused by the use of this software. Installing and using this software implies that you fully and unconditionally accept these terms. USE AT YOUR OWN RISK. Please fully test before using it.

Credits:

Credits go to Pablo Software Solutions for making (and further improving) their Extension Builder! And for continuously improving their WYSIWYG WB. Cudos!

We used the famous PHPMailer class as mailing routine, to make sure that everything will get along just fine. For more info on the PHPMailer class, please visit <http://phpmailer.worxware.com/index.php?pg=phpmailer>

Many thanks to the WWB Forum Users that have provided feedback! Without this feedback it wouldn't have been possible taking the script to this level.

Have fun!

George (Forum nickname Navaldesign)

Provider	SMTP Server	Server Secure Connection	Port
GMAIL	smtp.gmail.com	tls	587
YAHOO	smtp.mail.yahoo.com	no	25
HOTMAIL	smtp.live.com	tls	25
AOL	smtp.aol.com	no	25
LYKOS	smtp.mail.lykos.com	no	25