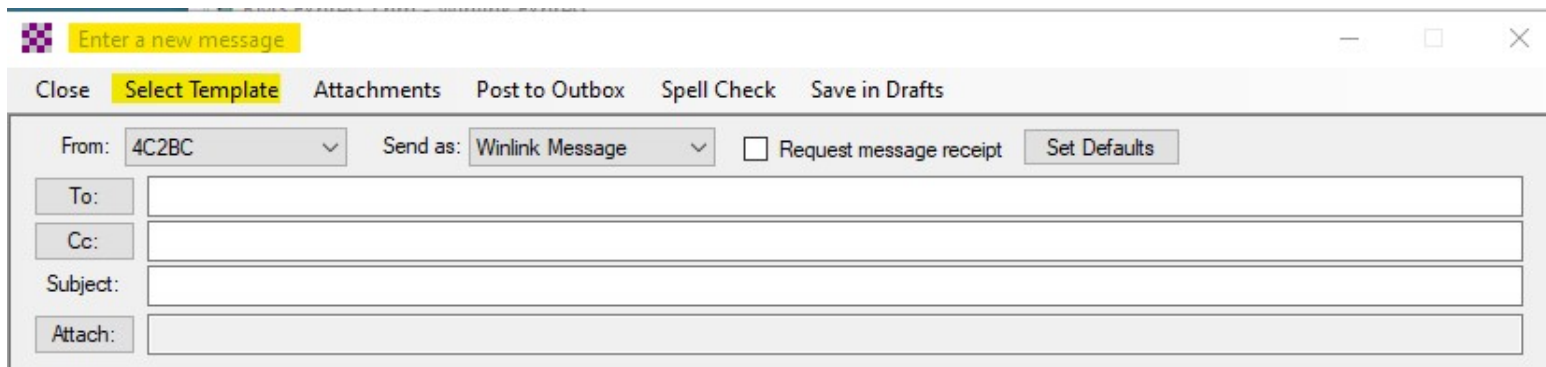


# Creating your own HTML Forms

**Note: Winlink Express has a series of pre-made HTML forms for your use. *These you can not modify.***



Winlink Express supports using html forms for requesting input from users. This is similar to using <ask...> and <select...> fields in templates, but it provides the full power of html including JavaScript, cascading style sheets, etc. Since Winlink Express is intended for field use without the Internet, forms must be stored on the client computer, and they should not reference items through the web.

Information entered through a form can be inserted into the message by using <var...> fields in a template. Optionally, the information can be packaged into an XML file and sent to the recipient. On the receiving end, Winlink Express will recognize that the XML file has form data, and it will merge it with an html file to present an attractive display.

Here is an outline of how to use forms with Winlink Express:

1. Create an html (web) form. You can create it using a text editor or a more powerful form generator program.
2. Store the html file(s) in the forms folder associated with a callsign. You can do this by copying the files into C:\RMS Express\callsign\Templates\ or by using the Forms Manager that you can access by clicking Messages/HTML Forms. Forms accessible to all Call signs are stored in C:\RMS Express\Global Folders\Templates\.
3. Create a template file that has a "Form:..." line listing the name of the form file. If you specify the form file name without a folder specification, Winlink Express will look for it in the Forms folder for the callsign. If you wish to store your forms in some other folder, enter the full file specification. Here is an example of a "Form:" line that would go in a template:

Form: Registration\_Entry.html

4. If you wish to have the data from the form packaged into an XML file and attached to the message, specify the name of the html file that should be used to display the data on the receiving end. Here is an example that tells the receiving Winlink Express to merge the form data with the html file named Registration\_Display.html when the received message is read. Note, you can create an XML file for transmission and also have the template insert data from the form in the body of the message.

Form: Registration\_Entry.html,Registration\_Display.html

5. If you wish to have a display form for the receiving end, you should create it and distribute to all potential recipients. They should store the form in their Winlink Express callsign\Forms\ folder.

6. If you want a specific template to be invoked when the receiving end creates a reply to a message created with this form, specify the ReplyTemplate command with the name of the template. For example:

ReplyTemplate: GenerateReply.txt

**The Windows service to display forms and accept input requires Administrator Privilege.** If you are running under a Windows account that doesn't have administrator privilege, you will get an "Access Denied" error. You can prevent that by right clicking on the Winlink Express icon on the desktop and selecting "Run as administrator" from the popup menu.

## Callsign-specific Forms, Global Forms and Standard Forms

Winlink Express supports three sets of forms: (1) **callsign-specific** forms that are associated with a specific callsign selected within Winlink Express; (2) **Global** forms that are available from any callsign; and (3) **Standard** forms that are created and supported by the Winlink Development Team. If you wish to create or edit forms, you should put your templates in the callsign-specific group or the global group. Forms in the standard template group should not be modified.

Callsign-specific forms are stored in a folder named "Templates" under the folder with the callsign within the Winlink Express program folder. For example, forms for W4PHS would be stored in C:\RMS Express\W4PHS\Templates.

Global forms are stored in the "Templates" folder under the "Global Folders" folder in the RMS Express program folder. For example, C:\RMS Express\Global Folders\Templates\

Standard forms are in the "Standard Templates" folder. For example, C:\RMS Express\Standard Templates\

## Creating a Form File for Winlink Express

You must follow some rules when creating an html form file for display by Winlink Express.

\* The form method must be POST, and you must specify "http://{FormServer}:{FormPort}" as the form action. The "{FormServer}" field will be replaced by the IP address of the Winlink Express form server. The "{FormPort}" field will be replaced by the IP port that the Winlink Express form server is using. By default, port 8001 is used, but you can change it on the Settings/Forms Setup screen. Here is an example of a form declaration:

```
<form id='Registration' method='post' action='http://{FormServer}:{FormPort}' enctype='multipart/form-data'>
```

\* If you're linking to a cascading style sheet in a separate file, you must specify the folder where the file is stored. If the .css file is in the same folder as the form, use the {FormFolder} insertion operator. Here's an example:

```
<link href="{FormFolder}FormStyle.css" rel="stylesheet" type="text/css" />
```

\* The Input declaration for the Submit button must use the POST method, and the value must be "Submit". Here are examples of Submit and Cancel buttons:

```
<input type='submit' name='Submit' value='Submit' method='Post' enctype='multipart/form-data' />
```

```
<input type='submit' name='Submit' value='Cancel' method='Post' enctype='multipart/form-data' />
```

\* Value insertion fields like <Callsign>, <DateTime> and <Position> can be used to insert values in forms. However, since the "<" and ">" characters have special meaning in html, you enclose the insertion keyword with curly braces, "{" and "}" rather than "<" and ">". For example: {Callsign}, {DateTime}, {Position}. You can use insertion operators to set default values for input fields. Here is an example:

```
<p>
Callsign: <input type='text' name='Callsign' value='{Callsign}'>
<p>
Time of incident: <input type='text' name='IncidentTime' value='{DateTime}' />
<p>
```

There are three special insertion fields for forms:

{FormFolder} -- Inserts the folder where the form is stored.

{FormServer} -- Inserts the IP address of the Winlink Express form server.

{FormPort} -- Inserts the IP port number Winlink Express is listening on for input from the form.

## Creating an HTML Display File for the Receiving End

If you wish to have an html file on the receiving end format and display information from a form sent in the XML file, you need to create the display file partner for the input form and distribute it to all potential recipients. Note: Input field values from the form are stored in compact form in XML format. Only the field values are transmitted, the form file itself is *not* transmitted and neither is the display html file. If the recipient doesn't have the display file required by the input form, the message will still be received and shown, but Winlink Express will not be able to format the data and display it using an attractive html file.

When Winlink Express opens a message, it checks to see if the message has an attached file providing values from a form. The file attachment name will be "RMS\_Express\_Form\_ *formname*.xml". If it finds a form data attachment, it first looks for the corresponding html display file in the \ *callsign* \Templates\ folder. If it doesn't find it there, it looks in C:\RMS Express\Global Folders\Templates\ . If the display file is found, it is opened, and any insertion fields in the html file are replaced by the values sent from the input form. Use {var *field\_name*} insertion fields to insert the form values. Here are examples:

```
Callsign: {var Callsign}
<p>
Time of incident: {var IncidentTime}
```

## Examples

Here is an example of a template that invokes a form named "IncidentInput.html" to collect data for an incident report. It specifies that a file named IncidentDisplay.html should be used to format the data on the receiving end. The data is packaged into an XML file named "IncidentDisplay.xml". Field values from the form are used by the template to insert data into the message body.

### Template -- Incident Report.txt

Form: IncidentReport.html,IncidentDisplay.html  
Subject: Incident report from <var Callsign> at <var IncidentTime> -- Severity: <var Severity>  
Msg:  
Callsign: <var Callsign>  
  
Time of incident: <var IncidentTime>

Severity: <var Severity>

Description: <var Description>

### Form -- IncidentReport.html

```
<head>
<link href="{FormFolder}FormStyle.css" rel="stylesheet" type="text/css" />
</head>

<h1><font color='red' Incident Report</font></font></h1>

<form class="form-container" id='IncidentReport' method='post' action='http://{FormServer}:{FormPort}' enctype='multipart/form-data'>
<div class="form-title"><h2>Incident Report</h2></div>
<div class="form-title">Callsign</div>
<input type='text' name='Callsign' value='{Callsign}'>
<p>
<div class="form-title">Time of incident</div>
<input type='text' name='IncidentTime' value='{DateTime}'>
<p>
<div class="form-title">Severity of incident</div>
<select name='Severity'>
  <option value="Normal report" selected>Normal report</option>
  <option value="Becoming dangerous">Becoming dangerous</option>
  <option value="Property damage">Property damage</option>
  <option value="Injured person">Injured person</option>
  <option value="Life threatening">Life threatening</option>
</select>
<p>
<div class="form-title">Description of incident</div>
<textarea name="Description" rows="6" cols="20"></textarea>
<br>
<p>
<input type='submit' name='Submit' value='Submit' method='Post' enctype='multipart/form-data' />
</p>
<p>
<input type='submit' name='Submit' value='Cancel' method='Post' enctype='multipart/form-data' />
</p>
</form>
```

### Cascading Style Sheet -- FormStyle.css

```
.form-container {
  border: 1px solid #f2e3d2;
  background: #c9b7a2;
  background: -webkit-gradient(linear, left top, left bottom, from(#f2e3d2), to(#c9b7a2));
  background: -webkit-linear-gradient(top, #f2e3d2, #c9b7a2);
  background: -moz-linear-gradient(top, #f2e3d2, #c9b7a2);
  background: -ms-linear-gradient(top, #f2e3d2, #c9b7a2);
  background: -o-linear-gradient(top, #f2e3d2, #c9b7a2);
  background-image: -ms-linear-gradient(top, #f2e3d2 0%, #c9b7a2 100%);
  -webkit-border-radius: 8px;
  -moz-border-radius: 8px;
  border-radius: 8px;
  -webkit-box-shadow: rgba(000,000,000,0.9) 0 1px 2px, inset rgba(255,255,255,0.4) 0 0px 0;
  -moz-box-shadow: rgba(000,000,000,0.9) 0 1px 2px, inset rgba(255,255,255,0.4) 0 0px 0;
```

```
box-shadow: rgba(000,000,000,0.9) 0 1px 2px, inset rgba(255,255,255,0.4) 0 0px 0;
font-family: 'Helvetica Neue',Helvetica,sans-serif;
text-decoration: none;
vertical-align: middle;
min-width:300px;
padding:20px;
width:300px;
}
.form-field {
border: 1px solid #c9b7a2;
background: #e4d5c3;
-webkit-border-radius: 4px;
-moz-border-radius: 4px;
border-radius: 4px;
color: #c9b7a2;
-webkit-box-shadow: rgba(255,255,255,0.4) 0 1px 0, inset rgba(000,000,000,0.7) 0 0px 0px;
-moz-box-shadow: rgba(255,255,255,0.4) 0 1px 0, inset rgba(000,000,000,0.7) 0 0px 0px;
box-shadow: rgba(255,255,255,0.4) 0 1px 0, inset rgba(000,000,000,0.7) 0 0px 0px;
padding:8px;
margin-bottom:20px;
width:280px;
}
.form-field:focus {
background: #fff;
color: #725129;
}
.form-container h2 {
text-shadow: #fdf2e4 0 1px 0;
font-size:18px;
margin: 0 0 10px 0;
font-weight:bold;
text-align:center;
}
.form-title {
margin-bottom:10px;
color: #725129;
text-shadow: #fdf2e4 0 1px 0;
}
.submit-container {
margin:8px 0;
text-align:right;
}
.submit-button {
border: 1px solid #447314;
background: #6aa436;
background: -webkit-gradient(linear, left top, left bottom, from(#8dc059), to(#6aa436));
background: -webkit-linear-gradient(top, #8dc059, #6aa436);
background: -moz-linear-gradient(top, #8dc059, #6aa436);
background: -ms-linear-gradient(top, #8dc059, #6aa436);
background: -o-linear-gradient(top, #8dc059, #6aa436);
background-image: -ms-linear-gradient(top, #8dc059 0%, #6aa436 100%);
-webkit-border-radius: 4px;
-moz-border-radius: 4px;
border-radius: 4px;
-webkit-box-shadow: rgba(255,255,255,0.4) 0 1px 0, inset rgba(255,255,255,0.4) 0 1px 0;
-moz-box-shadow: rgba(255,255,255,0.4) 0 1px 0, inset rgba(255,255,255,0.4) 0 1px 0;
box-shadow: rgba(255,255,255,0.4) 0 1px 0, inset rgba(255,255,255,0.4) 0 1px 0;
text-shadow: #addc7e 0 1px 0;
color: #31540c;
font-family: helvetica, serif;
padding: 8.5px 18px;
font-size: 14px;
text-decoration: none;
vertical-align: middle;
}
.submit-button:hover {
border: 1px solid #447314;
text-shadow: #31540c 0 1px 0;
background: #6aa436;
background: -webkit-gradient(linear, left top, left bottom, from(#8dc059), to(#6aa436));
background: -webkit-linear-gradient(top, #8dc059, #6aa436);
background: -moz-linear-gradient(top, #8dc059, #6aa436);
background: -ms-linear-gradient(top, #8dc059, #6aa436);
background: -o-linear-gradient(top, #8dc059, #6aa436);
background-image: -ms-linear-gradient(top, #8dc059 0%, #6aa436 100%);
```

```
color: #fff;
}
.submit-button:active {
text-shadow: #31540c 0 1px 0;
border: 1px solid #447314;
background: #8dc059;
background: -webkit-gradient(linear, left top, left bottom, from(#6aa436), to(#6aa436));
background: -webkit-linear-gradient(top, #6aa436, #8dc059);
background: -moz-linear-gradient(top, #6aa436, #8dc059);
background: -ms-linear-gradient(top, #6aa436, #8dc059);
background: -o-linear-gradient(top, #6aa436, #8dc059);
background-image: -ms-linear-gradient(top, #6aa436 0%, #8dc059 100%);
color: #fff;
}
```

Screen Display of Form

Incident Report

Callsign

W4PHS

Time of incident

2015-03-03 16:54:06

Severity of incident

Normal report

Description of incident

4 inches snow accumulation near I-65 and Concord Rd.

Submit

Cancel

When the fields are filled in and Select is clicked, the template fills in field in the message to be sent, and the XML file is attached to it:

Enter a new message

CloseSelect TemplateAttachmentsPost to OutboxSave in Drafts FolderSpell Check

From:W4PHS

☒ Winlink Message☐ Peerto-Peer Message☐ Request read receipt

To:

Cc:

Subject:Incident report from W4PHS at 2015-03-03 16:54:06 -- Severity: Normal report

Attach:RMS\_Express\_Form\_IncidentDisplay.xml;

Callsign: W4PHS

Time of incident: 2015-03-03 16:54:06

Severity: Normal report

Description: 4 inches snow accumulation near I-65 and Concord Rd.

```
<head>
<link href="{FormFolder}FormStyle.css" rel="stylesheet" type="text/css" />
</head>
<body class="form-container">
<h1>Incident Report</h1>
<div class="form-title"><h2>Incident Report</h2></div>
<p>
<div class="form-title">Callsign: {var Callsign}</div>
<p>
<div class="form-title">Time of incident: {var IncidentTime}</div>
<p>
<div class="form-title">Severity: {var Severity}</div>
<p>
  <div class="form-title">Description: {var Description}</div>
<p>
</body>
```

Screen Display of Received Message with Form Information

# Incident Report

## Incident Report

Callsign: W4PHS

Time of incident: 2015-03-03 16:54:06

Severity: Normal report

Description: 4 inches snow accumulation near I-65 and Concord Rd.