

OpenStore – Client file upload

OpenStore can allow the client to upload files, which are then visible in the Order Admin of the Manager. (Not the client Order Admin, although the templates can be changed to allow this)

The upload and attachment of the files to the order is done in 2 steps.

Step 1

The file is selected by the client and uploaded. This is done by the use of a file input.

```
<input id="cmdClientUpload" class="normalFileUpload" type="file" name="files[]" multiple>
```

When the files are selected a trigger is made to upload the files to the \Portals\#\NBSTemp folder and to populate a input field called “**optionfilelist**”.

Example:

```
function initClientFileUpload() {
    $('#input[id*="optionfilelist"]').val('');
    var filecount2 = 0;
    var filesdone2 = 0;
    $(function () {
        'use strict';
        var url = '/DesktopModules/NBright/NBrightBuy/XMLConnector.ashx?cmd=fileclientupload&itemid=' + $('#clientupload-productid').val();
        $('#cmdClientUpload').unbind();
        $('#cmdClientUpload').fileupload({
            url: url,
            maxFileSize: 5000000,
            acceptFileTypes: /(\\.|\\)(jpg|png|jpeg|pdf)$/i,
            dataType: 'json'
        }).prop('disabled', !$.support.fileInput).parent().addClass($.support.fileInput ? undefined : 'disabled')
        .bind('fileuploadprogressall', function (e, data) {
            var progress = parseInt(data.loaded / data.total * 100, 10);
            $('#progress .progress-bar').css('width', progress + '%');
        })
        .bind('fileuploadadd', function (e, data) {
            $.each(data.files, function (index, file) {
                $('#input[id*="optionfilelist"]').val($('#input[id*="optionfilelist"]').val() + file.name + ',');
                filesdone2 = filesdone2 + 1;
            });
        })
        .bind('fileuploadchange', function (e, data) {
            filecount2 = data.files.length;
            $('#processing').show();
        })
        .bind('fileuploaddrop', function (e, data) {
            filecount2 = data.files.length;
            $('#processing').show();
        })
        .bind('fileuploadstop', function (e) {
            if (filesdone2 == filecount2) {
                filesdone2 = 0;
                $('#processing').hide();
                $('#progress .progress-bar').css('width', '0');
            }
        })
    });
};
```

On upload the filename is encrypted using the store admin pin as a passkey. This is so the filename on the server is not known to a potential hacker.

Step 2

When the item is **added to the basket**, the “**optionfilelist**” field is used to process the uploaded files and attach them to the cart.

NOTES:

Client Files are moved from the temp folder to the “clientupload” folder, when added to the basket and are renamed to a random key name, so the files cannot be recognised.

The API command for client file upload is : “fileclientupload” (in the example the “itemid” param is passed in the url, this is not required)

The option is store settings for allowing client upload must be set for uploads to work.

There is an option in the product admin to activate the product client file upload option

There is no code on the “ClassicAjax” default templates to make uploading client files possible. This code will need to be added to the templates. You need the above code for html and JS put into the template. And a suitable test for the activated product upload option.

```
@{  
    // assign Model, so we can resolve var in VS  
    var product = (ProductData)Model.List.First();  
    var info = (NBrightInfo)product.Info;  
}  
  
@if (info.GetXmlPropertyBool("genxml/checkbox/chkfileupload"))  
{  
    <input id="cmdClientUpload" class="normalFileUpload" type="file" name="files[]" multiple>  
}
```

NOTE: Although precautions have been made to protect the upload from hackers, allowing uploading of files to your server can be VERY dangerous. Be aware of how to secure your server from external access.