

# MODULE 1: RULES CONFIGURATION

CONFIDENTIAL



# RULES CONFIGURATION



Manage rules for publishing, automation, validation and imports

## Objectives

Role of `post_input.aspx` and `filename.aspx`

Context

## Naming

- CMS Labels that appear in File View

- Published filenames and urls

## Validation

- Check for required fields

Filename/Url.aspx and post\_input.aspx template files control these rules

## POST INPUT CONFIGURATION



Create `post_input.aspx` by choosing New option

Make WYSIWYG a required field by adding

```
<% if (!context.InputForm.HasField("content"))  
    {  
        context.ValidationErrors.Add(  
            "content", "Please enter content.");  
    } %>
```

Edit your test file, remove the content from the WYSIWYG. CrownPeak will highlight the field

**Defaults** - Configure default filenaming conventions and extension

**FTP** - Define all servers available for publishing

**Package** - Link FTP to publishing group that can include other protocols

**Workflow** - Define the states when publishing can occur

**Publishing Properties** - Link files and folders to packages

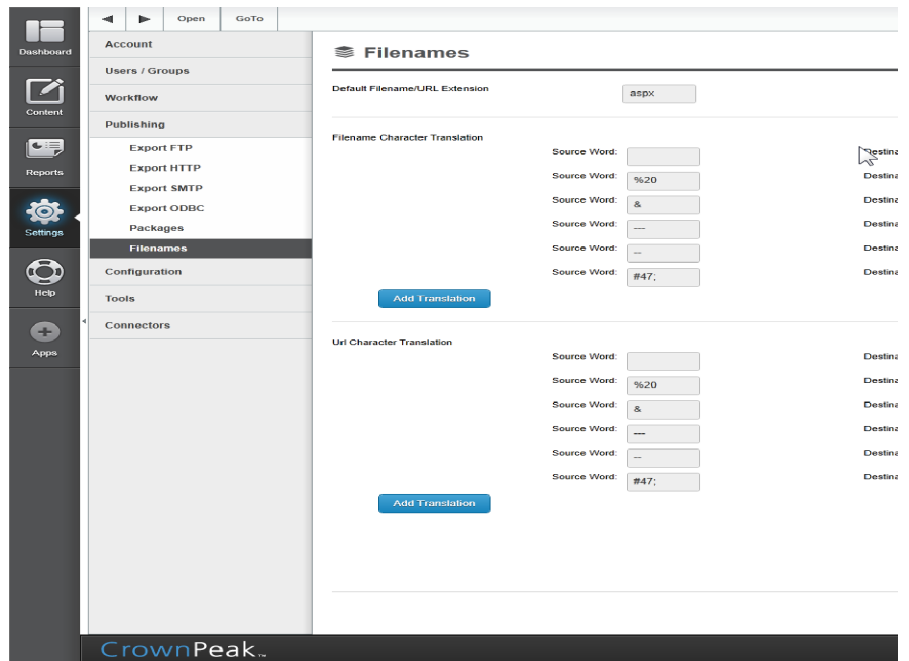
*All are required for publishing*

## Settings > Publishing > Filenames

Configure default extension

Configure default replacement characters

These setting can be overridden with Publishing Properties and filename/url.aspx configuration

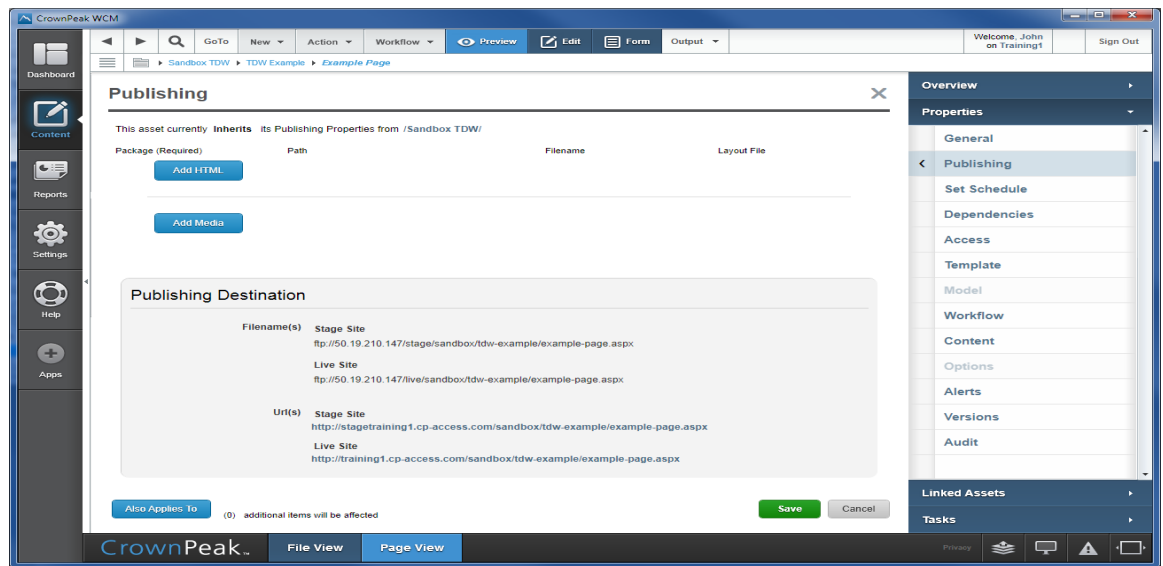


# CONFIGURE MODULE 1. CONFIGURE YOUR SITE

# DEFAULT FILENAMING CONFIGURATION



Go to test file and view General Properties. Note the url and filename



Add filename.aspx file to template

Add

```
<% context.PublishPath = "/test_" +  
context.RemotePath.Label + "." +  
context.RemotePath.Extension;  
%>
```

Go to General Properties, refresh and note the change



## Context

Methods, data, and properties particular to the environment, often determined by the active template file

Always refer to current context as simply "context".

You only need to worry about the type of context if you create functions that require context properties as parameters.

## CONTEXT EXAMPLES

The PostInputHelper class uses this to rename an asset to the title entered by the user.

In input.aspx, check as user's groups to show/add input fields

```
context.UserInfo.IsInGroup("CPTAdmin")
```

In output.aspx, OutputContext allows you to show and hide code based on publishing

```
context.IsPublishing
```

In filename.aspx, OutputContext allows you to set publishing path.

```
context.PublishPath = context.PropertiesFolder +  
context.RelativeFolder + asset.Id + ".html";
```

## EDIT FILENAMING CONFIGURATION



Replace code with library function

```
<% context.PublishPath =  
TDW.MakeFilename(context,  
asset); %>
```

Note the change

Type your question(s) in the question box.

# QUESTIONS

# TEST