



# Rules Configuration

- Manage rules for publishing, automation, validation and imports
- Objectives
  - Role of `post_input.aspx` and `filename.aspx`
  - Context

# Common rules configuration



- 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



1. Create post\_input.aspx by choosing New option
2. Make WYSIWYG a required field by adding

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

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

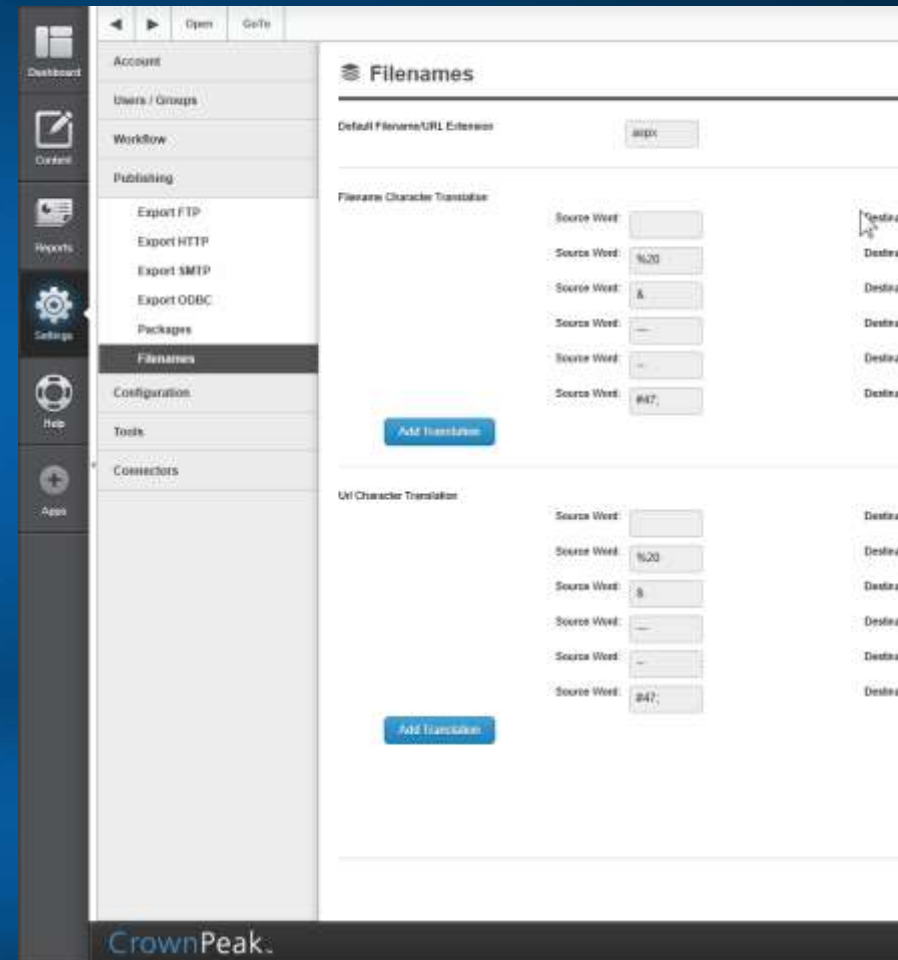
# Publishing Configuration: 5 steps



- **Defaults** - Configure default file naming 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***

# Defaults

- Settings > Publishing > Filenames
  - Configure default extension
  - Configure default replacement characters
  - These setting can be overridden with Publishing Properties and filename/url.aspx configuration



# Default Filenaming Configuration



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

**Publishing**

This asset currently inherits its Publishing Properties from /Sample Page TDW

Package Resource	Path	Filename	Local File
<a href="#">Add HTML</a>			
<a href="#">Add Media</a>			

**Publishing Destination**

Filename(s)	Stage Site	Live Site
	ftp://53.19.210.147/stage/andbox/ftp-example/sample-page.aspx	ftp://53.19.210.147/live/andbox/ftp-example/sample-page.aspx

URL	Stage Site	Live Site
	http://stage/andbox/ftp-example/sample-page.aspx	http://live/andbox/ftp-example/sample-page.aspx

[Add System to](#) (1) additional items will be affected

[Save](#) [Cancel](#)

# Edit Filenaming Configuration



1. Add filename.aspx file to template
2. Add

```
<% context.PublishPath = "/test_"  
+ context.RemotePath.Label + "."  
+ context.RemotePath.Extension;  
%>
```
3. Go to General Properties, refresh and note the change



# Context



- 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



1. Replace code with library function

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

2. Note the change

RULES CONFIGURATION

**TEST**

**QUESTIONS?**