



## Introduction To Models And Templates

# Introduction to Models and Templates

Introductions

Objectives

Agenda

**By the end of this training, you will be able to:**

- Explain the role of template files
- Identify types of templates
- Describe and create a microsite from a model

# Agenda

Introduction to Template Files

Project Overview

Types of Templates

Templates and Models

Building a Model

# Role of Template Files

## CMS allows us to

- Publish server-side code to run dynamic applications on an external web server
- Share design elements
- Establish global and/or default content
- Create configurations so that we can manage elements in a centralized place and reuse that content throughout the site(s)

# Leveraging the CMS

- Everything in the CMS is built from templates and assets.
- What can we do with templates and assets?
- First we need to understand what they are...



## What is a template?

- A named folder in the CMS (the template name) containing one or more template files
- Templates can create a 1 – many relationship with assets

## Default template files:

- Input.aspx – defining the input form
- Output.aspx – defining the code rendered when the template is previewed or published – the presentation

## What is an asset?

- Any file or folder in the CMS

## What does this mean?

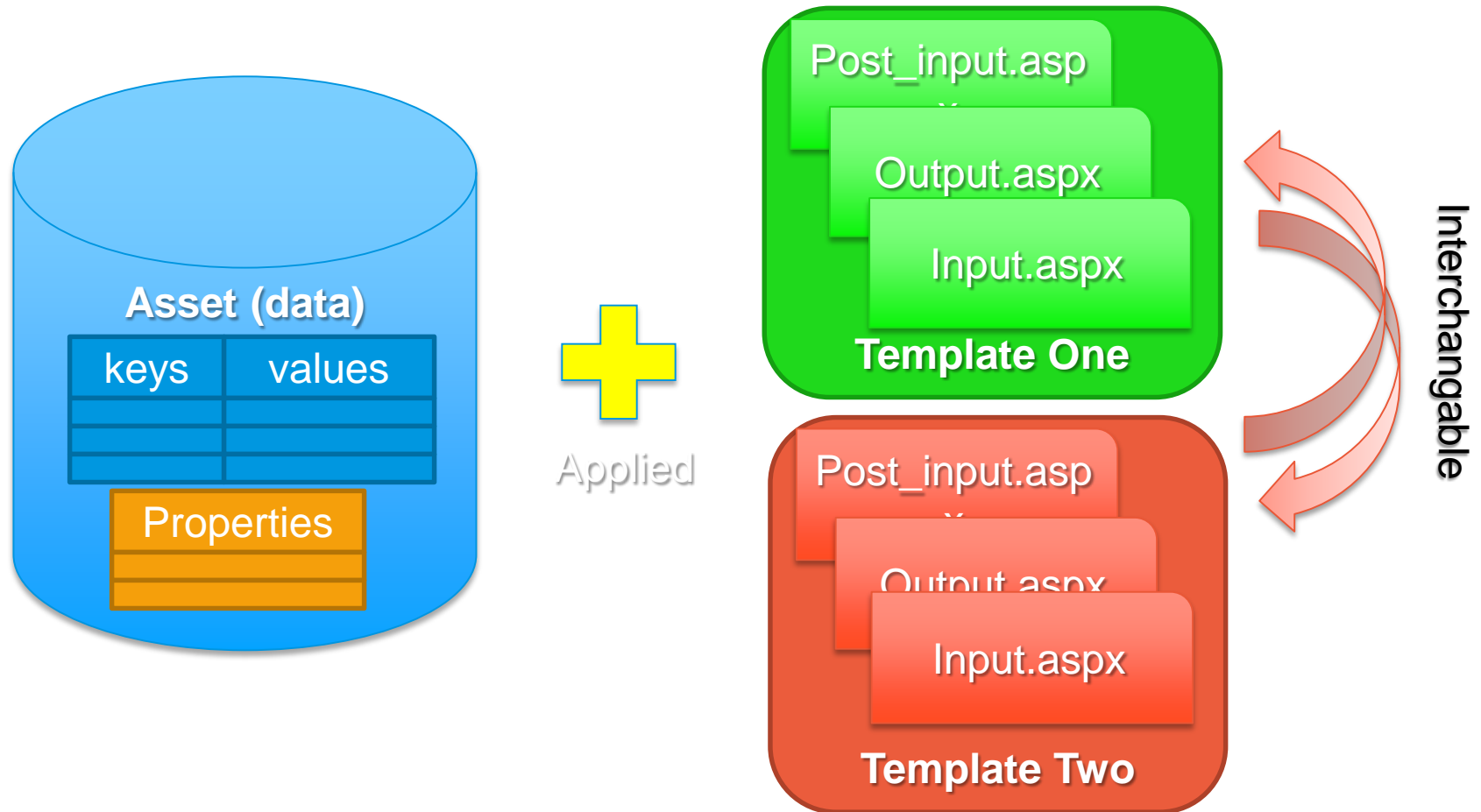
- Assets form site structure and store content/data
- Content/data is stored as key value pairs in each individual asset
  - i.e Key: page\_title -> Value: About Us data is stored in assets

Assets have properties and content

Assets have templates and workflow applied to them

- How do we get content into CrownPeak?
- Templates are applied to assets
- At the most basic level, users insert content into assets through the template's input file

# Content into CrownPeak



# Leveraging the CMS - Input

- Defines structure of form and data fields
- Defines interactivity
- Defines interactivity with content
- Developer decides on type and fields to create with content

## Users

- Experienced user can manage content with WYSIWYG fields
- Less Experience
  - More structured fields
  - Example: text field for header
  - Image field for images

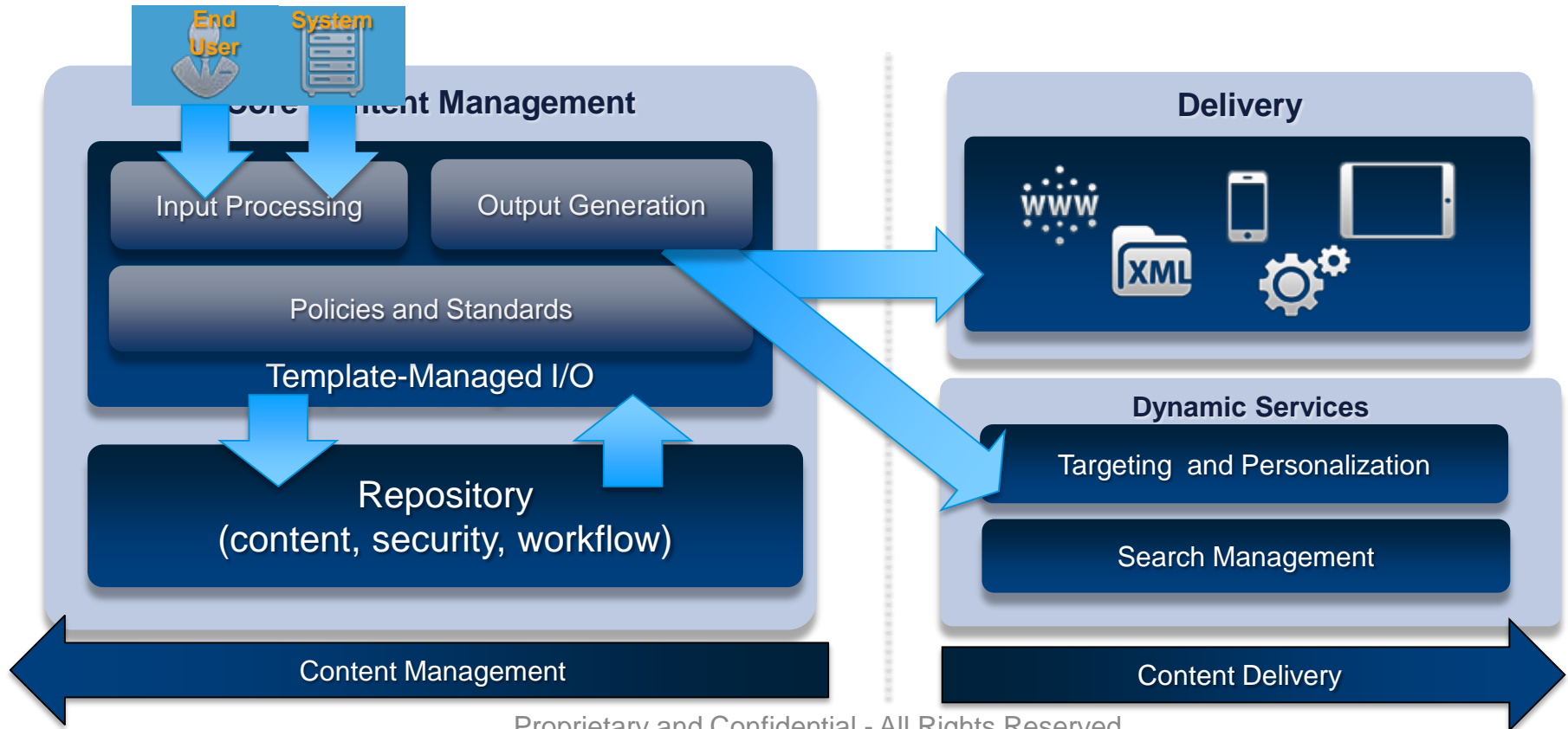
# Content out of CrownPeak

- How do we get content out of CrownPeak?
- Code for preview and publishing is defined in the template's output file
- Template can be changed without affecting the asset (no change in records or data)

- The CMS is your content database.
- Assets are your records, files, pages, etc.
  - There is no need for a separate content database for your web server
  - No database connections or SQL needed in your web site for content
- Content pages are rendered and published. You can limit dynamic components on the page to things that truly need to be dynamic, saving server resources and improving performance to the site visitor

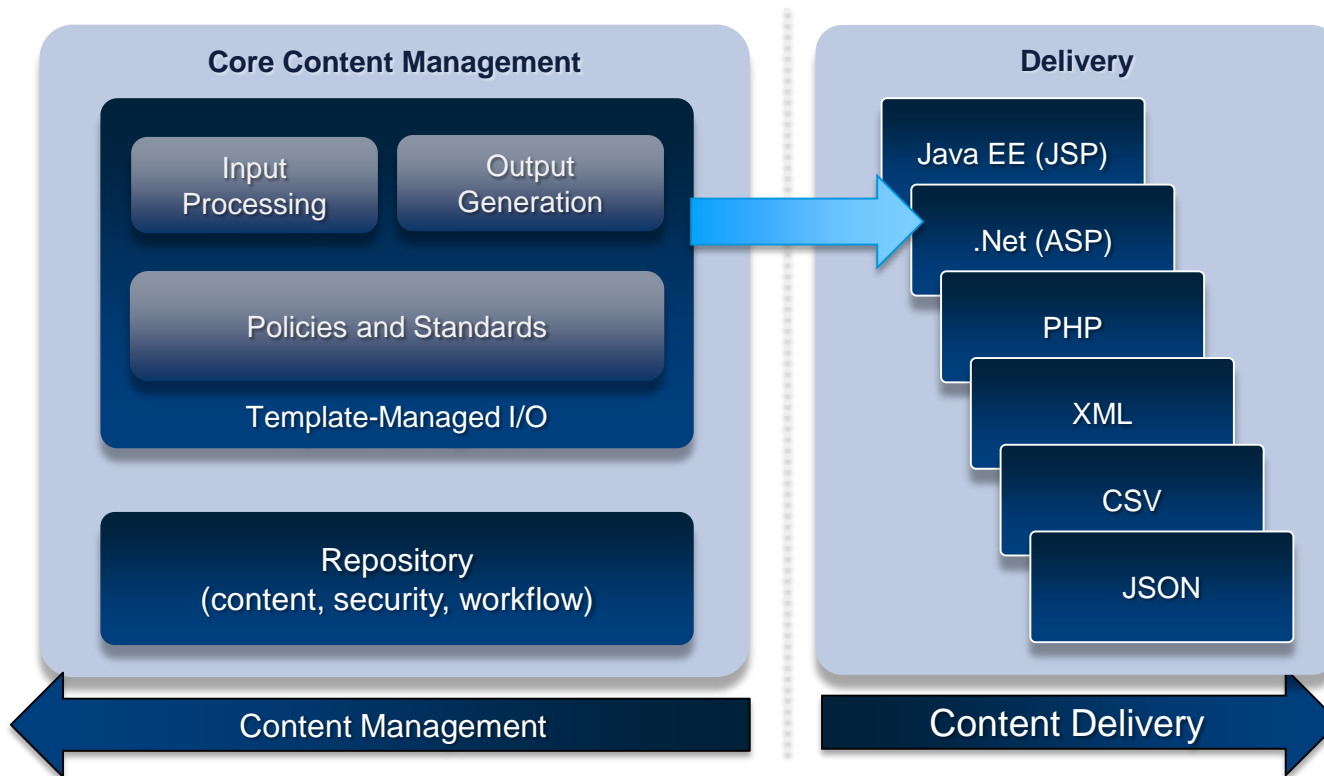


# CrownPeak Architecture



Proprietary and Confidential - All Rights Reserved

# Decoupled Deployment



Proprietary and Confidential - All Rights Reserved

## Two Examples Of Pages

- Static
- Dynamic

## Homepage –

Demo Site - v2 > Demo Site > US-EN > Index

## Template File

/Demo Site – v2/Demo Site Project/Templates/Home Page/

## Navigation Wrapper –

Demo Site – v2 > Demo Site Project > Templates > Nav Wrap

## Output

HTML Code, CrownPeak API Code

Surround <%%> (gator tags)

If site needs dynamic code:

- For example, server-side ASP.Net
- Output templates configured to deliver code with embedded server-side code
- Example

Press Release –

Demo Site - v2 > Demo Site > US-EN > Press Content > Press Releases > FDA  
Announces New Guidelines

Template File

/Demo Site – v2/Demo Site Project/Templates/Content Page

## Code Passes through...

Server-side code doesn't run in preview (it's not a web server)

It runs on the published page on the external web server (dev, stage, live)

If code conflicts with the CMS code it may need escaped out

Inline ASP.net and php can be entirely in the CMS output, and output can even set/change/render it

If you need compiled DLLs, manage your code solution out of the CMS and upload them in the CMS

You can create alternate previews for use in the CMS if needed

CrownPeak API code and `<$$>`.

`<$ $>` denotes embedded ASP.Net code

These tags (`<$$>`) will be converted to `<%%>` when published by CrownPeak



## Content Management Service

### Repository

Web Pages

Widgets

Configurations

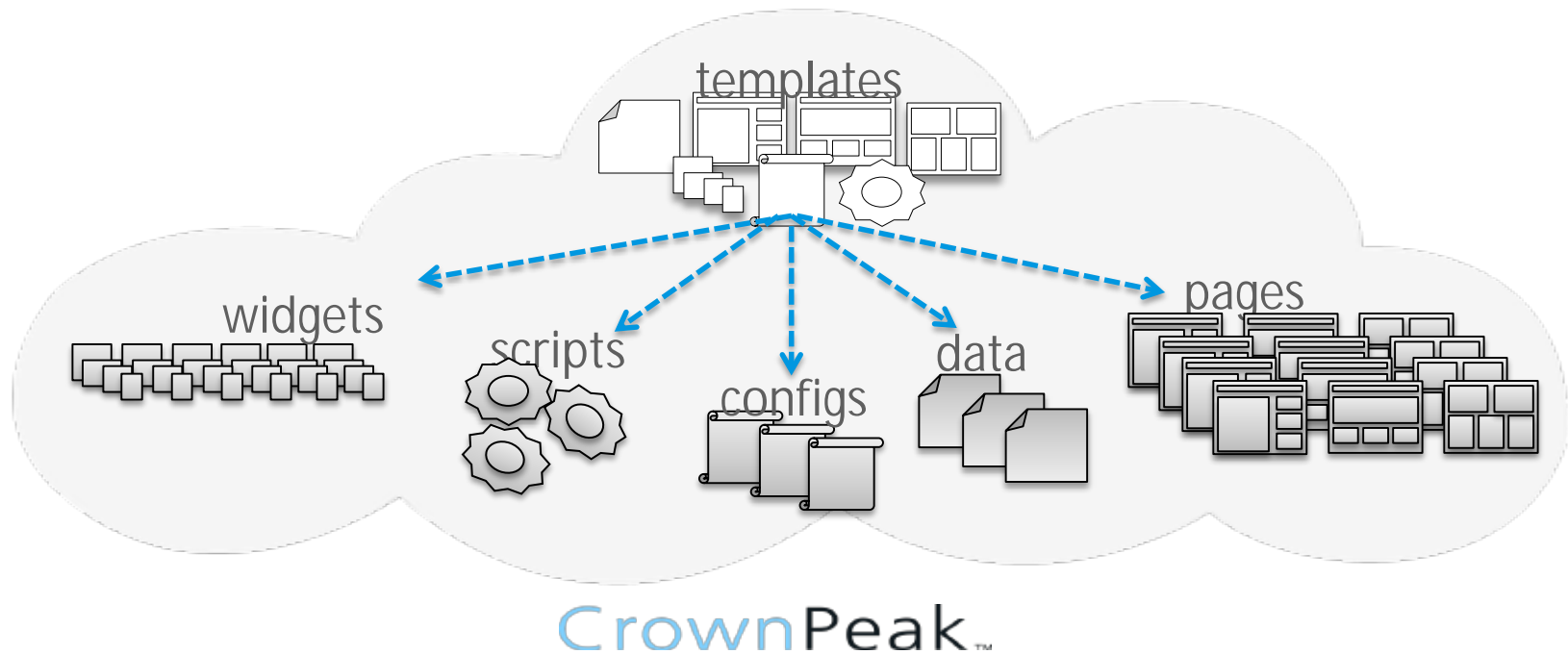
Data Files

Scripts

### Template-Managed I/O

Templates

# BASICS: TEMPLATE BUILDING BLOCKS



## TEMPLATE DEVELOPMENT STEPS

### 1. Input configuration

Controls the input form for content entry

### 2. Output configuration

Controls the preview and rendering of content for publishing

### 3. Rules configuration

Set up filenaming and field validation

### 4. Validate

Publish to web host, use test and improve user experience

### 5. Model configuration

Create **New** menu options for end users

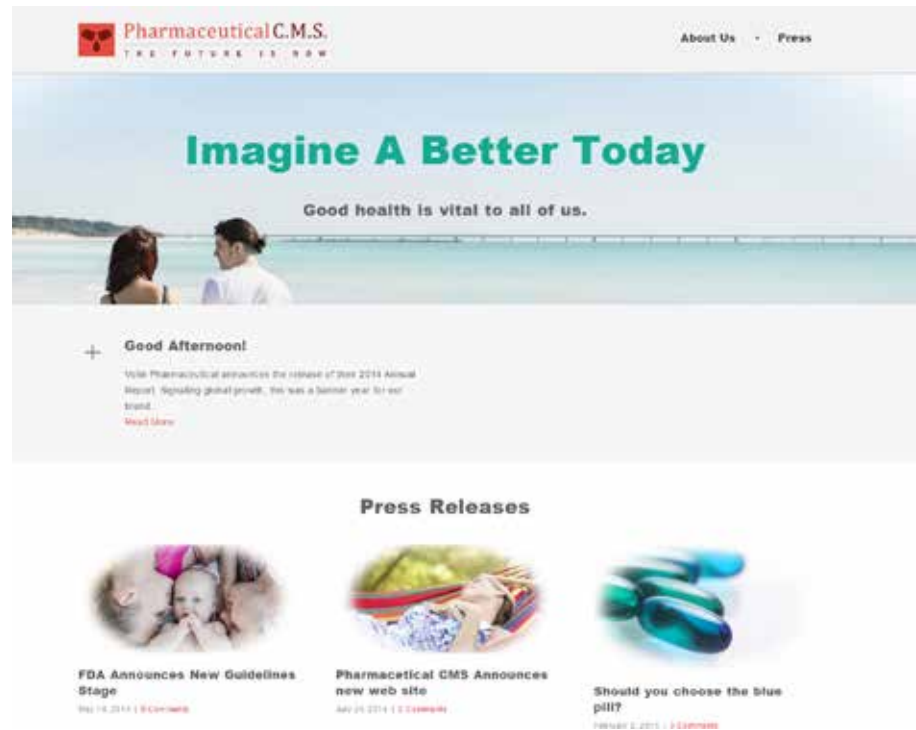
# Questions?



## Project Request

- Each Microsite will include
  - Homepage
  - Landing pages
  - Content pages

- **New Microsite**
  - requires a new set of templates and models
  - Leveraging an existing instance for workflow, user/group and publishing configuration
- **Templates and Models will drive creation of the microsites**
  - end user will be able to create as many microsites and customize as needed
- **This microsite structure should allow additional of functionality such as TMF and WCO to collect form data**



# CrownPeak HTML Guidelines

- HTML Delivery Requirements and Recommendation Guidelines can be found on Connect
- General Note: Design / Develop your HTML before you develop your CMS templates!



## Planning

- Review each page and output
- Determine requirements for templates
- Define groups, users, workflows, and access permissions.
- Define hosting and publishing

## Development

- Build Navigation Wrapper
- Build Configuration Templates
- Build Page Templates
- Build WCO, Search and 3<sup>rd</sup> party integrations

## End User Configuration

- Define Site Structure
- Modify default workflows
- Add users to default groups
- Configure Models

# BASICS: PLANNING YOUR MICROSITE

## Planning

- Review the page
- Determine requirements for template

## Development

- Build Page Template

## End User Configuration

- Configure Models

## Planning Templates and Functionality –

- User experience
- Hosting and Compliance Requirements

## Customers Provide:

- Sitemaps, functional specs, and wireframes
- Fully tested code for different page types
- Review these before you start development to save yourself headaches later

- Analyze the design to determine page types
- Receive client-tested and approved HTML
- Capture functional and technical requirements (UX, hosting, etc.)
- Plan Conversion of designs to templates
- Create Templates
- Create folder structure and any needed test assets
- Create models
  - Defines initial structure for new microsite and what types of additional pages/assets can be created within it
- Assign models in CMS to folders
  - Ensures that Users will create microsities in appropriate places

## Three Page Designs

- Homepage
- Press Release Index Page
- Content Page

## Three Page Designs

- Homepage
- Press Release Index Page
- Press Content Page

## Determine Three Templates

- Homepage
- Sorted Index Page
- Content Page

**But that's not all...**

**Next we'll discuss...**

- Leveraging the CMS
- Ways you can use templates beyond the obvious

# Questions?





# Template Types: An Overview

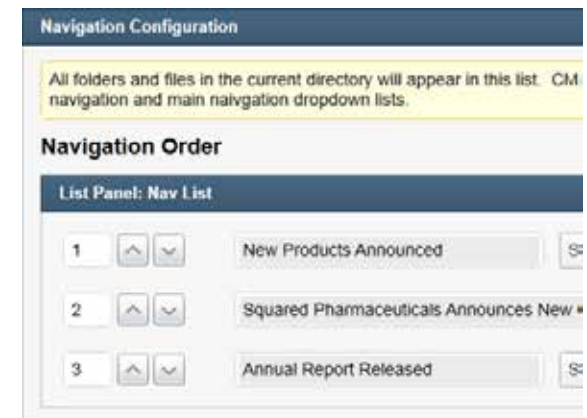
- **Page Templates in CrownPeak include the following template files:**
  - `input.aspx`: controls form to enter content
  - `output.aspx`: controls layout that is published and previewed
  - `post_input.aspx`: validates entered content to make sure it fits the rules
  - `filename.aspx`: sets name of page (or pages) when published

# Other Uses for Templates?

- Templates are used for pages, but can also be used to store content and retrieve it when building other pages...
- **Configuration Files**
  - Users enter content (input template file)
  - Template code
    - Developers grab and use content from external assets
  - Users enter content – no preview
  - Content used by other templates to display shared information ((navigation, logo, etc.) in multiple pages

# Configuration Templates

- Used for Global, Site, and Section Configuration Files
- Enable Users to Manage Shared Content (headers, footers, menus, and global images, navigation, settings, etc.)
- Configuration templates include:
  - `input.aspx`: form to enter content
  - `post_input.aspx`: process content
    - Combining multiple fields into one – easier for user
    - Post processing – easier for developer



## Best Practice

Places shared content in page templates

## Wrapper templates Include:

output.aspx: includes Doctype, html, head tag, css and javascript references, placeholder so it can be reused in multiple page templates

Used for shared content that is inserted into multiple pages

## Templates Include:

input.aspx: form to enter content

output.aspx: layout that is published, usually just part of a page

Preview - may not render with styles unless a preview.aspx is created

Publishing - Render on page where widget is selected or publish as a usercontrol (.net) or SSI (ASP and PHP)

post\_input.aspx:

content validation and post-processing

filename.aspx: sets name of file when published

If the widget will not publish a file - set filename to blank

# Developer Templates

Used for css, javascript and code assets

The asset IS the template, and there is only an output

Allows you to edit/version/branch front end code and use the CMS api to build code and create references to images and other assets

Filenames are controlled by filename files in the System folder

Code attached and applied in CrownPeak

## Templates Include:

*{named as needed}.aspx*:

Email.aspx is an example of one that is available in CMS

Allows users to script email notification in workflow

Developers add scripts as needed and link to particular workflow step using execute file option



## Similar to configuration template type

### Best Practice

Enable users to manage content used for categorization, such as keywords, author names, site sections

Use these templates to manage shared content, build drop-down and check-box values, etc.

### Templates Include:

input.aspx: form to enter content.  
usually the name and description  
post\_input.aspx: process content  
checking for duplicate categories

## Templates with non-html output

### Best Practice

Enables users to export XML, JSON, RSS and other formats for use in content feeds and other applications

For templates aggregating content from multiple sources, or for existing templates with an alternate output.

### Templates Include:

output-xml.aspx – just a renamed output.aspx.

filename.aspx – customizes file extension for designer data output (.xml, .rss, etc.)

```
<?xml version="1.0"?>
<Content>
  <PageId>3270</PageId>
  <PageTitle><cpinline data-cp-field-asset="3270" data-cp-field-
name="page_title" data-cp-field-type="textbox">About
Us</cpinline></PageTitle>
  <PageContent><![CDATA[<p>AdventGeneral is a diversified financial
services firm headquartered in Chicago, with locations across the
globe. We are a private company made up of approximately 1,200
employees, of which 25% have ownership in the firm.</p>
<p>We aim to understand, anticipate and meet our clients' changing
financial needs with a multitude of high-quality products and services.
Since our founding in 1937, AdventGeneral has grown to become:</p>
<ul type="square">
<li><strong>A multiproduct money manager of assets for institutions as
well as investment advisory for private clients</strong>, with
expertise in hedge fund strategies, private equity, currency
management, institutional real estate, equities and fixed income</li>
<li><strong>One of the largest broker/dealers </strong>for institutions
and other financial entities</li>
<li><strong>One of the largest independent insurance consultants and
among the top 25 in the nation</strong>, with more than $3.5 billion in
premium volume</li>
<li><strong>An international leader in consulting services</strong>,
offering a broad range of financial advisory capabilities, compensation
and executive benefit strategies, as well as real estate consulting</li>
</ul>
]]></PageContent>
</Content>
```

## Templates used in the CMS for custom reporting on assets

### Templates Include:

output.aspx

report data that can  
be viewed in CMS

input.aspx

form to enter content. Usually  
report options or  
parameters

post\_input.aspx

post-process and  
validate content

post-publish.aspx

For adjusting report parameters after publishing

The screenshot shows a web interface for a CMS report. The title is "Release 84429 Report". Below the title, there is a "Description" section with the following details:

- Status: **Approved**
- Status Date: 2/4/2014 7:40:49 PM (Date the status was changed)
- From: [Empty]
- To: **Stage**

Below the description is a table with the following columns: Asset Id, Branch Of, CMS Path/Publishing Paths, Created, Status, Last Modified, Published Live Date\*, and Published By\*.

Asset Id	Branch Of	CMS Path/Publishing Paths	Created	Status	Last Modified	Published Live Date*	Published By*
84030		site-logo.gif <a href="#">/Acesandbox1/CrownPeak Training Sample/_Assets/img/site-logo.gif</a>	2/3/2014 3:48:46 PM	Authoring	2/3/2014 3:49:18 PM		A. W.
84033		Content Page Example <a href="#">/Acesandbox1/CrownPeak Training Sample/Content/Content Page Example</a>	2/3/2014 3:55:09 PM	Authoring	2/3/2014 3:56:41 PM		C.
82906		Global Config <a href="#">/Acesandbox1/CrownPeak Training Sample/_Global/Global Config</a>	2/1/2014 12:09:35 AM	Draft	2/3/2014 4:11:40 PM		G.
83907		Master Page <a href="#">/Acesandbox1/CrownPeak Training Sample/_Global/Master Page</a>	2/1/2014 12:09:36 AM	Draft	2/1/2014 12:09:36 AM		N.

\* Captured on publish.

# Import Templates

Usually based on a page template

Enables users to automate content creation / management

Configured to import content via FTP (polled by CMS) or SMTP (emailed to CMS)

Once created behave like other assets and need approval to be published and viewed on live site

## These templates include:

### **input.aspx**

form to enter /edit content

### **output.aspx**

layout that is published and previewed

### **post\_input.aspx**

validates entered content to make sure it fits the rules

### **filename.aspx**

sets name of page (or pages) when published

### **ftp/smtp\_import.aspx**

script that runs on imported data. This where developer sets the rules for asset creation and modification

Advanced Page Templates that gather information from other assets for display

**Index pages** and **Landing Pages** listing recent updates or allowing navigation to specific pages

**Homepages** showing highlights and linking to different parts of a site

Composite pages grabbing content from multiple assets and displaying in a pre-set way

**These templates include:**

**input.aspx** form to enter content/set options

**output.aspx** layout that is published and previewed, often use advanced API functions such as filters to automate queries and load content from other assets in the system

# Search Templates

Search templates are page templates that have usually have CrownPeak search queries hard-coded into their output

Some options may be in input but search code/query is generally managed by CrownPeak

## C# TEMPLATE CODE

### Basic Guidelines

- Templates use C# syntax, but are not creating .net projects in the CMS
- CrownPeak's C# API is not a library for use on .NET sites or a web service
- Developers can use C# functions and custom objects and classes



## C# API LIBRARIES

There are 3 libraries available for C# templates

**CMS API** – This is the core API library and contains methods and functions for creating template files and adding rules for processing.

**Custom Library** – This is a local library that is maintained and developed in each CMS. It allows CMS template developers to create their own class files. Library files are available from /System/Libraries/ or in your project.

**Services** – This is a best practices library (in development). As CrownPeak Professional services develops classes, the best of breed will be added to the Services library for reuse across all CrownPeak Platform instances.

Details on objects and methods and best practices are available in Development Center

<http://developer.crownpeak.com/>

TITLE AND  
CONFIGURE  
CONTENT  
YOUR SITE  
LAYOUT

# Questions?



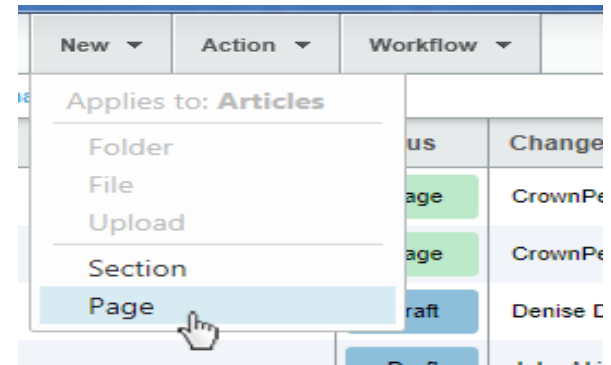
# Templates and Models

# Microsite Model

- Blueprint for content creation
- Enable users to create single page, site selection, or entire site
  - Depends on configuration

- **Models**


- Accessed from the New menu
- Implemented by developers or Admins, models are used for creation of most pages and sections
- Models ensure that all configuration settings will be the same
  - Template
  - Workflow
  - Access Properties
- Can be used to create a single asset, a section, or an entire site
- Ideal for
  - End-users, templated content, simple or complex structures, creating brand-new content



- Structure for site
- Using these properties, developer uses model to define site structure
- Defines what user can create and where
- If model is site selection or site
  - Will include all files, folders, sub-folders
  - Need to be available when structure is created



## Simple Model

Single asset or single folder: e.g., new Article or new Press Release Year

System > Models > Sandbox > Denise with Solutionary > Simple Model							
Type	Label	Status	Changed By	Changed On	Size	Template	ID
	Simple Model	Draft	Denise Duncan	11/1/2012 9:26:56 AM	0 B	Detail Page	4723

## Intermediate Model

Model with a group of assets or folders

System > Models > Sandbox > Denise with Solutionary > Intermediate Model							
Type	Label	Status	Changed By	Changed On	Size	Template	ID
	Intermediate Folder		Denise Duncan	11/1/2012 9:32:12 AM	1		4728
	Intermediate Model	Draft	Denise Duncan	11/1/2012 9:30:24 AM	0 B	Detail Page	4727



## Recursive Model

Folder that calls itself as a model – allows for deep hierarchy of content  
 E.g., a Subsection folder that can create new pages or new subsections inside of it to create nested subsections

The screenshot shows a content management system interface. At the top, there is a breadcrumb trail: System > Models > Sandbox > Denise with Solutionary > Recursive Model > Recursive Folder. Below this is a table with the following data:

Type	Label	Status	Changed By	Changed On	Size	Template	ID
	Index	Draft	Denise Duncan	11/1/2012 9:39:09 AM	0 B	Detail Page	4737

Below the table, there is a 'New' dropdown menu open, showing the following options: Folder, File, Upload, Recursive Folder, and Recursive Model. The 'Recursive Folder' option is highlighted. The dropdown also shows 'Applied to: Recursive Example' and a small table with the following data:

Type	Label	By	Changed On	Size	Template	ID
	This is a folder	Duncan	11/1/2012 9:41:22 AM	2		4739

# Three Model Types

- **Models are configured in the System Folder**
  - /System/Models/
- **Best Practice**
  - Create a separate folder for the models of each site

- **Steps to Create Model**
  - Create container for Microsite Model
  - Create containers for
    - Assets
    - Sections
  - In each container, build the options the user will see in **New Menu**
- **Assign Model**
  - Go to CMS – add model to CMS location to create microsite

- **Create Model for Microsite**
  - /Demo Site - v2/Demo Site/
  - Show how it contains folders, subfolders, etc.
  - Explain how users can add content pages
- **Demo**
  - Create new microsite

# Questions?



Thank you  
for attending this  
training!