



Input Configuration

- For content entry
- Objectives
 - Identify fields
 - Create input template
 - Review the Input class and what it lets us do

TEST: Input fields

Text Box

This is a help message

1 [Navigation Bar]

2 [Text Column]

3 [Image]

4 [Contact List]

WYSIWYG

Edit View Insert Format Tools Table Form Help

This is help text for the WYSIWYG

List Panel: Panel

Name

Telephone

Name

Telephone

Image

No item selected.

Select Clear

This is help text for the image

Global Content

Page Content

TEST: Input fields

1

Text Box

This is a help message

2

WYSIWYG

Edit View Insert Format Tools Table Form Help

Normal Paragraph | Arial, Helvetica, sans-s... | 12px

Paragraph | content

Design Code

This is help text for the WYSIWYG

1

Template Development Workshop Example

3



2

consectetur adipiscing elit. Nulla non odio lectus. Cras nec justo in libero imperdiet fermentum nec sed elit. Vestibulum vel feiis sollicitudin metus. Mauris posuere. Maecenas eget ipsum enim. Donec viverra ultricies tellus, in mattis libero sodales nec. Curabitur in tellus erat. Phasellus interdum, velit in facilisis tristique, magna enim varius lorem, id consequat nunc mauris vitae risus. Morbi eget commodo mauris. Curabitur eget est enim porttitor. Sed lorem risus, aemper in congue eget, ornare at lac adipiscing lorem. Donec ut nunc mi, vel pellentesque lorem. Cras in libero in tellus ornare tempor quis quis nibh. Sed vitae ligula quam, vel rhoncus velit.

4

Contact One
212 400 1388

Contact Two
212 400 1389

Contact Three
212 400 1390

4

List Panel: Panel

Name

Telephone

Name

Telephone

3

Image

No item selected.

Select Clear

This is help text for the image



Page Content

CrownPeak API: Key Concepts

- Template Code
 - Templates use C#
 - 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
- Objects
 - Objects allow access to CMS database to add and edit content and variables
 - CMS abstracts database layer, no SQL statements are needed.
 - Adding fields in template creates variables, stores values
 - `Input.ShowTextBox("Page Title", "page_title");`
 - Developer can access / manipulate content with methods such as
 - `Asset articlesFolder = Asset.Load("/Site/Articles/");`
 - `List<Asset> myArticles = articlesFolder.GetFilesList();`

CrownPeak API: Key Concepts

- Details on objects and methods available in API reference
 - <http://help.crownpeak.com/CMSAPI/>
- CMSAPI Objects
 - **Asset** – User-entered data, Links, access content in multiple assets
 - **Context** – Methods, data, and properties particular to the environment, often determined by the active template file
 - **Panel, PanelEntry** – Sub-assets either generated from a list of assets (as might be used in an index page) or from a collection of fields (such as a user-defined list of related links for a page)
 - Additional object types: Images, uploaded files, etc.
 - All object types listed in CMSAPI documentation under `Crownpeak.CMSAPI Namespace`

Create Template



1. In CDC, go to `/System/Templates/TDW/` directory.
2. Right-click and select `New > Template C#`. Name it - TDW Template
3. Double-click `input.aspx` to open Visual Studio
(By default, it will include `input.aspx` and `output.aspx` template files.)

Create Input fields



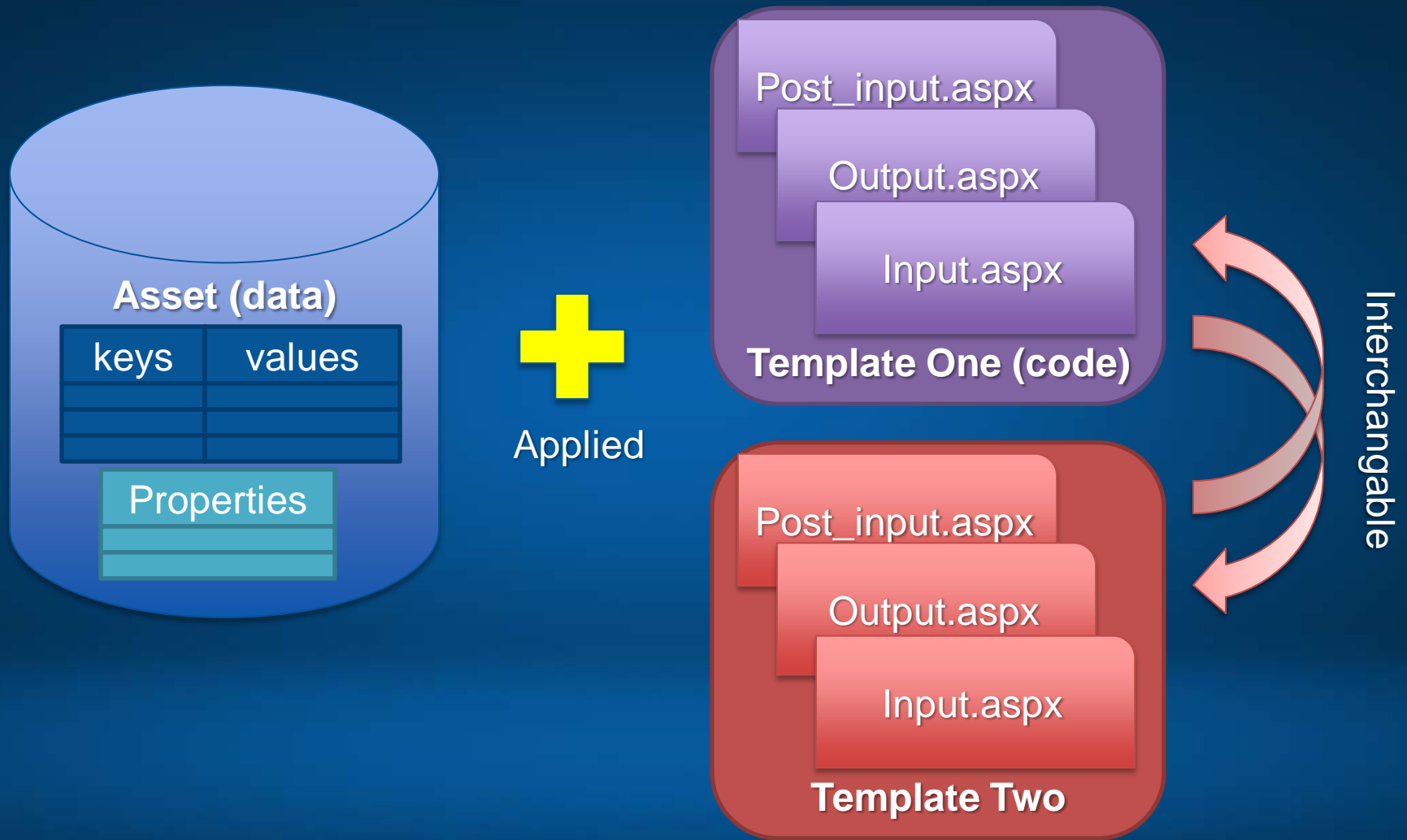
1. Create a Title field –
`Input.ShowTextBox()`
2. Create an Image upload field –
`Input.ShowAcquireImage()`
3. Create a WYSIWYG field –
`Input.ShowWysiwyg()`
4. Create a panel –
`while(Input.NextPanel()){}`

Create Test File



1. Navigate to /Sandbox TDW/ and create a folder
2. Create a new file and select /System/Templates/TDW/TDW Template/
3. Set Access properties to inherit
4. Set Workflow to “Basic Workflow”
5. Edit file to test input and enter content
6. Save and View content from Content properties

Assets and Templates



Refining Your Input Form



1. Separate Your input into Tabs

–

```
Input.StartTabbedPanel()
```

```
...Input.NextTabbedPanel()
```

```
...Input.EndTabbedPanel()
```

2. Create a Message, used for end user notes or instructions

– `Input.ShowMessage()`

3. Remove the old panel fields and add a widget selection panel to the second tab –

```
Input.ShowAcquireDocument()
```

Refining Your Input Form



4. Use a library function to create meta fields on the third tab – `InputHelper.ShowMetaInput()`
5. Review the API documentation for other field types...
 - Some are for input fields (Checkboxes, Dropdowns, ShowDate),
 - others are for presentation only (e.g., Tabs, Headers, ExpandPanels).
 - Some are both (e.g., DropDownContainer)

QUESTIONS?