Draft Boilerplate Texts for Plug-in’s README.md and Wiki

Overview

In preparation for the release of v2.0 of the Adapt framework, documentation of plug-ins is being improved. Goals include greater standardization of layout and providing more complete information. The most significant benefit will be a better end-user experience, but it is hoped that the improvements will lead to documentation that is easier to maintain and to expand.

The target of these proposals is the group of core plug-ins. It is hoped that these will serve as a model for the maintainers of non-core plug-ins.

READMEs, Wikis, and audiences

GitHub automatically builds a README.md file when initializing a repository. It is viewable upon entering the repository. It is widely used. GitHub also creates a wiki. A link to it is viewable upon entering the repository. It is much less widely used than the README.

The README is customarily one page. It is typically cloned alongside source code. A GitHub wiki can have multiple pages with more prominent navigation. It doesn’t travel with the source code.

It has become accepted that Adapt users fall into two broad categories: technical user and non-technical end user. It is acknowledged that technical skill is a range, and the information sought by the two groups often overlaps.

What is being proposed?

It is proposed that the README and wiki of all core plug-ins be built-out using templates in time for the release of v2.0 framework. The templates assume an audience and depth of information as listed below.
README:

- reader has a technical background
- plug-in is primarily used with framework
- reader has a preference for brevity

Wiki:

- reader is a non-technical end user
- plug-in is primarily used with the authoring tool
- reader has a preference for depth of information

Wireframes of the proposed README and its associated wiki can be found in RDME_wiki_wireframes.pdf. Using boilerplate text wherever possible ensures consistency and reduces the time it takes to implement the template. The rest of this document is focused on the proposed boilerplate texts.

Draft Boilerplate Texts

Introduction

- These draft texts reference the major sections found in the wireframes of RDME_wiki_wireframes.pdf.
- The draft texts presented here are intended for use with core plug-ins. While it is expected that these models may find use with non-core plug-ins, this document does not account for them.

1. Draft Texts for README.md

Usage

Boilerplate R1:

[Core Plug-in] is a [*presentation component*] bundled with the Adapt framework.

Notes:

- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- [presentation component] is to be replaced by the most appropriate plug-in type:
  - presentation component
  - question component
  - extension
  - menu
  - theme
- Adapt framework is a link to the framework README.md

Boilerplate R2:

Visit the [**Core Plug-in**] wiki for more information about its functionality and for explanations of properties.

Notes:

- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- Visit [**Core Plug-in**] wiki is a link to the plug-in’s wiki home page

Installation

Boilerplate R3:

As one of Adapt's *core components*, [**Core Plug-in**] is included with the installation of the Adapt framework and the installation of the Adapt authoring tool.

Notes:

- core components is a link to adapt_framework/wiki/Core-Plug-ins-in-the-Adapt-Learning-Framework
- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- installation of the Adapt framework is a link to adapt_framework/wiki/Setting-up-your-development-environment
- installation of the Adapt authoring tool is a link to adapt_authoring/wiki/Installing-Adapt-Origin

Boilerplate R4:

If [**Core Plug-in**] has been uninstalled from the Adapt framework, it may be reinstalled. With the Adapt CLI installed, run the following from the command line:

```
adapt install [adapt-core-plugin]
```

Alternatively, this [component] can also be installed by adding the following line of code to the adapt.json file:

```
"[adapt-core-plugin]": "*"
```

Then running the command:

```
 adapt install
```

(This second method will reinstall all plug-ins listed in adapt.json.)

Notes:

- This is the first of two bullets addressing installation; one for the framework, the other for the authoring tool.
- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- Adapt CLI is a link to adaptlearning/adapt-cli
- `[adapt-core-plugin]` is to be replaced with the repository name (e.g., adapt-pageLevelProgress)
- [component] is to be replaced by the most appropriate plug-in type:
    - component
    - extension
    - menu
    - theme

Boilerplate R5:

If [**Core Plug-in**] has been uninstalled from the Adapt authoring tool, it may be reinstalled using the Plug-in Manager.

Notes:

- This is the second of two bullets addressing installation; one for the framework, the other for the authoring tool.
- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- Plug-in Manager is a link to adapt_authoring/wiki/Plugin-Manager

Settings Overview

Boilerplate R6:

The attributes listed below are used in [components.json] to configure [**Core Plug-in**] and are properly formatted as JSON in example.json. Visit the [**Core Plug-in**] wiki for more information about how they appear in the authoring tool.

Notes:

- [components.json] to change appropriately for the plug-in type
- "attributes" is used for consistency with the wiki page named "core model attributes."
- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- example.json is a link to the plug-in's example.json
- **Core Plugin** wiki is a link to plug-in's wiki page anchored at "Settings"

Boilerplate R7:

core model attributes: These are inherited by every Adapt component. Read more.

Notes:

- This text is to be used only with components. It does not apply to other plug-in types.
- **core model attributes** is a link to adapt_framework/wiki/Core-model-attributes
- **Read more** is a link to adapt_framework/wiki/Core-model-attributes
- This sentence and its links are a substitute for explanations of title, displayTitle, body, and instructions. It is purely a way to save space. Brevity is preferred in the README (and some components have many attributes).

Note: Boilerplate text is proposed in Appendix A for many commonly used attributes. These can be used if applicable to the plug-in.

Limitations

Boilerplate R8:

No known limitations.

Notes:

- This text only applies to plug-ins with no significant limitations.
- Including the Limitations section when the plug-in has none demonstrates the issue was addressed and not merely overlooked.

Browser Specs

Boilerplate R9:

This [component] has been tested to the standard Adapt browser specification.

Notes:

- This text only applies to plug-ins that have met the specification standards.
- [component] is to be replaced by the most appropriate plug-in type:
  - component
  - extension
  - menu
  - theme
- standard Adapt browser specification is a link to where the browser specification is documented.

Metadata

Boilerplate R10:

Version number:
Framework compatibility:
Author / maintainer: Adapt Core Team
Accessibility support: WAI AA
RTL support: Yes
Cross-platform coverage:

Notes:

- Expected value for **Version number**: 1.1.6 (The current version of the plug-in.)
- Expected values for **Framework compatibility**: 1.1.1, 1.1.0, 1.0.0 (Framework version numbers.)
- Expected values for **Author /maintainer**: Adapt Core Team (Core plug-ins are maintained by Adapt.)
- Expected values for **Accessibility support:** WAI AA (Or other accessibility standards that apply.)
- Expected values for **RTL support**: Yes
- Expected values for **Cross-platform coverage**: (?)

2. Draft Texts for Wiki

Custom Sidebar

Boilerplate W1:

Usage
Installation
Settings
Limitations
Browser Specs

Notes:

- All are linked to the sections of the same name, except Usage which links to the top of the page.

Usage

Boilerplate W2:

[**Core Plug-in**] is a [*presentation component*] bundled with the Adapt framework. It is included in the Adapt authoring tool.

Notes:

- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- [presentation component] is to be replaced by the most appropriate plug-in type:
  - presentation component
  - question component
  - extension
  - menu
  - theme
- Adapt framework is a link to the framework README.md
- Adapt authoring tool is a link to the authoring tool wiki.

Installation

Boilerplate W3:

As one of Adapt's *core components*, [**Core Plug-in**] is included with the installation of the Adapt framework and the installation of the Adapt authoring tool.

Notes:

- core components is a link to adapt_framework/wiki/Core-Plug-ins-in-the-Adapt-Learning-Framework
- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- installation of the Adapt framework is a link to adapt_framework/wiki/Setting-up-your-development-environment
- installation of the Adapt authoring tool is a link to adapt_authoring/wiki/Installing-Adapt-Origin

Boilerplate W4:

If [**Core Plug-in**] has been uninstalled from the Adapt framework, it may be reinstalled. With the Adapt CLI installed, run the following from the command line:

```
adapt install [adapt-core-plugin]
```

Alternatively, this [component] can also be installed by adding the following line of code to the adapt.json file:

```
"[adapt-core-plugin]": "*"
```

Then running the command:

```
 adapt install
```

(This second method will reinstall all plug-ins listed in adapt.json.)

Notes:

- This is the first of two bullets addressing installation; one for the framework, the other for the authoring tool.
- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- Adapt CLI is a link to adaptlearning/adapt-cli
- `[adapt-core-plugin]` is to be replaced with the repository name (e.g., adapt-pageLevelProgress)
- [component] is to be replaced by the most appropriate plug-in type:
    - component
    - extension
    - menu
    - theme

Boilerplate W5:

If [**Core Plug-in**] has been uninstalled from the Adapt authoring tool, it may be reinstalled using the Plug-in Manager.

Notes:

- This is the second of two bullets addressing installation; one for the framework, the other for the authoring tool.
- [Core Plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)
- Plug-in Manager is a link to adapt_authoring/wiki/Plugin-Manager

Settings

Boilerplate W6:

The settings listed below are the properties that can be configured within the authoring tool. Listed first is the name of the property as it is presented in the authoring tool. It is followed in parentheses by the property as it is named in the [core plug-in] source code. Finally, an explanation is provided to help the course author better utilize the property.

Notes:

- Simply an explanation of how the properties are laid-out.
- [core plug-in] is to be replaced with the common name (e.g., Page Level Progress), not the repository name (e.g., adapt-pageLevelProgress)

Boilerplate W7:

The relationship between the [component]'s properties and the authoring tool's interface is governed by properties.schema.

Notes:

- Offers a justification for why the properties are laid out as they are and why only certain properties will be listed.
- [component] is to be replaced by the most appropriate plug-in type:
    - component
    - extension
    - menu
    - theme
- properties.schema is a link to the plug-in's properties.schema.

Boilerplate W8:

Title (`title`): A reference title for the component. Title is distinct from the Display Title which, if present, appears above the component. If Display Title is rather long, Title provides the opportunity to use a shortened form in tighter spaces, such as in menus or in the drawer.

Display Title (`displayTitle`): Optional text that will display as a title above the component. It can be used as a headline. In the authoring tool, Title and Display Title may be linked for easy input. Click the chain icon to toggle linking.

Body (`body`): Optional text that appears above the component. It can be used in a variety of ways, including as an introduction to the content in the component. HTML is permitted.

Notes:

- This text is to be used only with components. Typically it would not apply to other plug-in types.

Boilerplate W9:

Further settings for this component are:

Instruction (`instruction`): This optional text appears above the component. It is frequently used to guide the learner's interaction with the component.

Classes (`_classes`): Custom classes predefined in the theme's Less files may be applied to this component. To do this, enter the name of the CSS class. If using more than one class, separate each with a space.

Layout (`_layout`): This defines the horizontal position of the component in the block that contains it. Choose from `left`, `right`, or `full`. When `left` or `right` is selected, another component can occupy the space next to it. If `full` is selected, the component spans the width of the containing block.

Notes:

- This text is to be used only with components. Typically it would not apply to other plug-in types.

Note: Boilerplate text is proposed in Appendix B for many commonly used attributes. These can be used if applicable to the plug-in

Limitations

Boilerplate W10:

No known limitations.

Notes:

- This text only applies to plug-ins with no significant limitations.
- Including the Limitations section when the plug-in has none demonstrates the issue was addressed and not merely overlooked.

Browser Specs

Boilerplate W11:

This [component] has been tested to the standard Adapt browser specification.

Notes:

- This text only applies to plug-ins that have met the specification standards.
- [component] is to be replaced by the most appropriate plug-in type:
  - component
  - extension
  - menu
  - theme
- standard Adapt browser specification is a link to where the browser specification is documented.

Appendix A

README.md: Draft Texts for Attributes

_component (string): This value must be [core plug-in short name].

_classes (string): CSS class name to be applied to [core plug-in]'s containing `div`. The class must be defined in one of the Less files. Separate multiple classes with a space.

_layout (string): This defines the horizontal position of the component in the block. Values can be `full`, `left` or `right`.

_items (array): Multiple items may be created. Each item represents one [plug-in specific noun] and contains values for [list of object properties].

_graphic (object): The image that appears [location]. It contains values for [list of object properties].

alt (string): This text becomes the image's `alt` attribute.

src (string): File name (including path) of the image. Path should be relative to the src folder (e.g., `course/en/images/origami-menu-two.jpg`).

title (string): This text becomes the image's `title` attribute.

large (string): File name (including path) of the image used with large device width. Path should be relative to the src folder (e.g., `course/en/images/origami-menu-two.jpg`).

small (string): File name (including path) of the image used with small device width. Path should be relative to the src folder (e.g., `course/en/images/origami-menu-two.jpg`).

attribution (string): This text is displayed as a caption or credit for the image. HTML is permitted.

placeholder (string): This text is displayed as input before a learner interacts with the element.

_attempts (integer): This specifies the number of times a learner is allowed to submit an answer. The default is [integer].

_isRandom (boolean): Setting this value to `true` will cause the `_items` to appear in a random order each time the component is loaded. The default is [boolean].

_feedback (object): These various texts will be displayed depending on the submitted answer. `_feedback` contains values for three types of answers: `_incorrect`, `_partlyCorrect`, and `correct`.

correct (string): Text that will be displayed when the submitted answer is correct.

_incorrect (object): Texts that will be displayed when the submitted answer is incorrect. It contains values that are displayed under differing conditions: `final` and `notFinal`.

_partlyCorrect (object): Texts that will be displayed when the submitted answer is partially correct. It contains values that are displayed under differing conditions: `final` and `notFinal`.

notFinal (string): Text that will be displayed when the submitted answer is incorrect while more attempts are permitted.

final (string): Text that will be displayed when the submitted answer is incorrect but no more attempts are permitted.

_buttons (object): Multiple button objects and properties that govern their use.

_submit (object): A button object that triggers the evaluation of a response. Contains values for `buttonText` and `ariaLabel`.

_reset (object): A button object that clears the learner's selection or input. Contains values for `buttonText` and `ariaLabel`.

_showCorrectAnswer (object): A button object that displays the correct answer. Contains values for `buttonText` and `ariaLabel`.

_hideCorrectAnswer (object): A button object that removes the correct answer from view. The opposite behavior of `_showCorrectAnswer`. Contains values for `buttonText` and `ariaLabel`.

_showFeedback (object): A button object that displays the current feedback text. Contains values for `buttonText` and `ariaLabel`.

buttonText (string): Text that is used to label the button.

ariaLabel (string): This text labels an element that has no visible text label. The text is read aloud by screen readers.

disabledAriaLabel (string): This text labels a disabled ("greyed-out") element that has no visible text label. The text is read aloud by screen readers.

remainingAttemptsText (string): This text is concatenated to the number of times a learner may submit an answer.

remainingAttemptText (string): This text is displayed when the learner is permitted to submit an answer once.

Appendix B

Wiki: Draft Texts for Attributes

Graphic: (`_graphic`)
Alternative text: (`alt`): This text becomes the image's `alt` attribute. It is intended to be a text equivalent of the information conveyed by the image. It is read by assistive technology such as screen readers.

Select graphic: (`src`): File name (including path) of the image. This text will be set when you upload an image or select one from the Asset Manager.

Graphic title: (`title`): This text becomes the image's `title` attribute. It should complement the `alt` attribute. It is displayed in a "tool tip" when the cursor hovers over the image.

Buttons: (`_buttons`)
Hide Correct Answer: (`_hideCorrectAnswer`)
Button Label text to hide the model answer: (`buttonText`): This text will appear on the button that removes the correct answer from view. For example, "My Answer".

Show Correct Answer: (`_hideCorrectAnswer`)
Button Label text to show the model answer: (`buttonText`): This text will appear on the button that displays the correct answer. For example, "Correct Answer".

Reset: (`_reset`)
Button Label text for the reset button: (`buttonText`): This text will appear on the button that clears the learner's selection or input and returns the component to its original. For example, "Reset".

Submit: (`_submit`)
Button Label text for the submit button: (`buttonText`): This text will appear on the button that triggers evaluation of the learner's response. For example, "Submit Your Answer".

Number of attempts: (`_attempts`): This number indicates how many times a learner will be allowed to respond to a question in an effort to answer correctly.

Random order: (`_isRandom`): Select `true` or `false` to specify whether questions are presented to the learner in a random order.

Question weight: (`_questionWeight`): This number specifies the proportional weight used in calculating the learner's score.

Feedback: (`_feedback`): Feedback is provided when the Tutor extension (adapt-contrib-tutor) is enabled.
Feedback for correct answer: (`correct`): This text will display when the learner submits the correct response.

Feedback for incorrect answers: (`_incorrect`)
Incorrect final answer: (`final`): This text will display when the learner submits an incorrect response on the last permissible attempt.

Incorrect non final answer: (`notFinal`): This text will display when the learner submits an incorrect response with additional attempts remaining.

Feedback when answer is partly correct: (`_partlyCorrect`)
Partially correct final answer: (`final`): This text will display when the learner submits a response on the last permissible attempt, and the response is not wholly correct.

Partially correct non final answer: (`notFinal`): This text will display when the learner submits response with additional attempts remaining, and the response is not wholly correct.