impression new project
manager's workbook

impression

table of contents

the impression new project manager's workbook

This page intentionally left blank

introduction 1

workbook overview

Creating content in Impression requires two tools, the Content Creation Tool (CCT) and the Runtime Engine (RTE). The Impression New Project Manager's Workbook provides you with an approach to determining what your RTE will look like and to modifying the CCT to match it so that creating and editing your content is easy. You will need to consider what types of functionality and appearance your project requires. The project requirements as well as runtime editors and feature limitations will help guide these content decisions. At the completion of this document, you will have a set of project guidelines that you can give your programmers and media artists so they can create the content that fits your specifications.

The method presented in this workbook will help limit the production process to the features needed for your particular project. Remember that even if you have not decided on all of the specifics of your project (for example, determining how the runtime will function), making decisions about storyboard types, backgrounds, and supported media types can help you save time and money and avoid frustration. Even if these features may need to change later in the creation process, it will be easy to reset or modify project properties once they are in place.

In some instances, you will specify options common to both the CCT and RTE. In these cases, the storyboard data in the CCT is similar to that of the runtime. Customizing the CCT to accept only data supported in the runtime ensures that the CCT visuals match those in the runtime. This will create consistency and reduce confusion during the content creation process.

Chapter 3 defines the runtime options, face placement and style, interface functionality, content delivery, and CCT options for each storyboard type. Each subsection provides a brief overview of the storyboard type, how you can customize it to fit your particular needs, and how it can help you select the features that are best for your project. You will find a series of fields for option selection and definitions; enter the appropriate information to establish the values for each aspect of your project. The sections marked "for further exploration" help guide you on features that go above and beyond the normal scope of the CCT. In some cases, they can key you to, and help define, additional capabilities not implemented in the CCT that the runtime shell can execute.

Use the CCT to edit the project properties dialog available from the Groups Window's Tools menu. You can access and modify options like supported storyboard types, background and graphic properties, supported media types, and storyboard fields in Project Properties.

Once these properties are set, you can build the individual storyboards using the various fields and specific storyboard features to create lessons. Once you have created the content, you will also use the lesson/group hierarchies to organize and manage lessons.

With respect to the CCT, you will need to decide which features to support. You should limit features to the types of storyboards, storyboard options, and CCT options that your project must support. Choosing these options early will reduce the confusion of those creating your courseware since they will not have access to prohibited fields and features. Also, consider the type of media and lesson/group organization required. It is helpful for the media storage and lesson organization to mirror the other's content—so, if hierarchical lessons are supported, you should also arrange the media assets this way.

You will also need to specify the runtime options. Runtime functionality affects the playing, viewing, and packaging of content, so carefully considering the options is vital. Decide on the RTE "look and feel" options like screen size, appearance, and shell commands. Decide what data the runtime will handle here and which deployment type you will use.

This page intentionally left blank

one-minute worksheet 2

The One-Minute Worksheet provides a quick snapshot of the workbook's content and format. This single page will give you an idea of the available CCT and RTE options and the order of their presentation. The following will help you focus on the big picture, but we recommend that you complete each detail field in the subsequent chapters to customize the CCT and RTE to your project's specifications. Filling in the workbook's contents will allow you to map supported features and decide on your project's look and performance.

storyboard type and type-specific options

◦ Supported storyboard types:

◦ Storyboard-type options:

◦ General tab options:

◦ Supoorted remediation types:

CCT-specific options

◦ Media directory organization:

◦ Project content (lesson/group) organization:

RTE-specific options

◦ Runtime look and feel:

◦ Controlling content flow:

◦ Runtime deployment options:

This page intentionally left blank

storyboard types and options 3

In this chapter, you will choose and define the storyboard type-specific options necessary to establish the style and behavior of your project. You will use the various fields to customize the RTE and CCT to your specifications.

rendering surfaces vs. faces

The Impression Runtime Engine Base Class Framework includes individual player classes for most storyboard types (multiple storyboard types are supported with the GenericPlayer class). There are two player classes: those that handle the drawing of storyboard visuals and those that do not.

Player classes that draw the storyboard data directly are called "rendering surface players." They take a rendering surface (a Flash MovieClip) and draw (or render) the storyboard content onto it. Classes with a rendering surface include the Canvas player, the Flash video and Flash object players, and the Rich Text player. Each class allows a degree of customization through a matching [player]Format object, which contains the data used to modify the display.

Player classes that do not draw the storyboard data directly are "face players." These players require the RTE programmer to develop a visual display called a *face* to display the data. Classes that require faces include the Multiple-Choice, Matching, and Menu classes as well as the GenericPlayer, which is for simple storyboard types that do not require interaction. The appearance and behavior of face-based classes are entirely up to the programmer.

face placement and style

Face placement and style, together, determine how storyboards look and behave in the runtime. Onscreen element location, size, appearance, and interaction with students are face placement and style factors. General and type-specific storyboard options, which determine what fields and options are available to content creators, are another important style and placement factor.

Think about what kinds of visual content for storyboards and remediation to support. Based on how you want the project to look and act, select only those storyboards and storyboard options that support this interaction type with the runtime. Remember to turn off features that you are not supporting in the CCT using the Project Properties available from the Groups Window's Tools menu.

Why are content creators tasked with face placement and style? For starters, it is impossible to define every style possibility because of the number of storyboard and feature combinations. Secondly, placement and style limitations might restrict feature capabilities.

This page intentionally left blank

storyboard types

Select the storyboard types your project supports.

- ☐ Title
  A screen with a fixed visual appearance used to introduce students to the lesson content
- ☐ Summary
  Reinforces and wraps up the lesson and can also be used to present aggregate information such as student scoring
- ☐ Rich Text
  Storyboard for building and editing large amounts of text
- ☐ Canvas
  A still graphic-based storyboard that can be assigned elements and interaction
- ☐ Flash video
  Plays traditional and stepped .SWF Flash animations
- ☐ Flash object
  Embedded Flash .SWF file designed for one- or two-way communication with the RTE
- ☐ Placeholder
  Common screens that do not change from lesson to lesson within a course, such as a copyright or disclaimer
- ☐ Menu
  Used as an organization and navigation tool, especially when developing lessons with nonlinear navigation
- ☐ Multiple-Choice Text
  Multiple-choice question format with one correct text answer and up to three text distractors
- ☐ Multiple-Choice Graphics
  Multiple-choice question format with one correct graphical answer and up to three graphical distractors
- ☐ Matching Text to Text
  Matching question format with directions, four text question/answer pairs, and one text distractor
- ☐ Matching Text to Graphics
  Matching question format with directions, four text question/graphical answer pairs, and one graphical distractor
- ☐ Container
  Used as an aggregation and navigation aid when developing courseware with a nonlinear learning pathway

title

The Title is a face-based storyboard that introduces the lesson. It is typically the first storyboard of the lesson.

runtime options

Specify the appearance of the Title face in the space below.

This page intentionally left blank

CCT options

Use the fields below to support and modify the CCT options associated the Title storyboard type.

☐ Supports additional text

If supporting additional text, describe its purpose in the space below.

summary

The Summary is a face-based storyboard that wraps up lesson content. It is typically the last storyboard in a lesson or lesson section. Additional uses for the Summary storyboard include aggregating test score data.

runtime options

Specify the appearance of the Summary face in the space below.

“for further exploration”

Summary storyboards often display aggregate information, including test scores. Tokens can be embedded in Summary storyboards and later replaced by specific detokenized data. Decide what additional text and functionality is needed.

If additional Summary storyboard functionality is supported, describe it in the space below.

This page intentionally left blank

rich text

The Rich Text storyboard is a rendered-surface type where text is the primary feature. Rich Text is versatile, with possible uses ranging from welcome screens and objective lists to warning screens and test score displays.

runtime options

Use the runtime options to set the defaults for text type and appearance.

Fill in the default TextFormat object values in the spaces provided.

Text styles *(margins, style, etc.)*
(This is a Flash TextFormat option.): _____

Background graphic opacity %: _____

CCT options

Use the fields below to specify the CCT options associated with the Rich Text storyboard and editor.

Editor size (width x height): _____

Default font *(name and size)*: _____

☐ Supports background graphic

☐ Allows user to change font name

☐ Allows user to change font size

This page intentionally left blank

canvas

Canvas storyboards display a single still graphic, called the *background*. Content creators can layer additional shapes, text, or graphic elements on top of the background graphic. Elements can be assigned one of several different action types. The action occurs during user interaction with the element. Shape and text elements can have associated system-defined styles based on their actions. The Canvas player is a rendering surface, which means that the player itself handles all of the visual drawings.

runtime options

You can define the visual style of the elements, general background options like background size and color, and behavioral types in the RTE. The changes made in the runtime options will affect the CCT's settings.

Use the spaces below to set the default properties associated with the RTE. Specify options for background size, color, shape, and text style. Also, decide if the CCT will support custom styles.

general/background options

Canvas size (width x height): _____

Background color/opacity %: _____

Autosize background graphic *(y/n)*: _____

general element styles

Default stroke width: _____

Default stroke color: _____

Default text color: _____

Default text font/style/size: _____

Allow custom styles *(y/n)*: _____

action-specific styles

Assign action-specific styles to Canvas hotspots (interactive elements) on the storyboard. You must set style properties for each action type and for the possible action state, usually active, inactive, or initialized.

The following entries define the shape and text element color, as well as the pixel width of the shape elements, for the specified action type and action state.

Jump to storyboard *(destination uninitialized)*: _____

Jump to storyboard *(destination initialized)*: _____

Jump to storyboard *(destination complete)*: _____

Pop-up text *(not shown)*: _____

Pop-up text *(active)*: _____

Pop-up text *(shown)*: _____

Pop-up graphic *(not shown)*: _____

Pop-up graphic *(active)*: _____

Pop-up graphic *(shown)*: _____

Play sound *(not played)*: _____

Play sound *(played)*: _____

Hyperlink *(not clicked/visited)*: _____

Hyperlink *(clicked/visited)*: _____

Command *(not executed)*: _____

Command *(executed)*: _____

This page intentionally left blank

action-specific behaviors

Action-specific behaviors control how the student initiates and completes interaction with elements.

*Choose only **one** option for each of the following action behavior types:*

click type

- ☐ OnPress
  Actions begin when the user presses the mouse button over the element.
- ☐ OnRelease
  Actions begin when the user presses, then releases, the mouse button over the element.

pop-up behavior

- ☐ Press and hold
  Pop-ups begin when the user presses the mouse over the element and disappear when the user releases the mouse.
- ☐ Click on/click off
  Pop-ups begin when the user clicks (presses or releases) the mouse over the element and disappear when the user clicks on another element or on the background.
- ☐ Mouseover
  Pop-ups begin when the mouse moves over the element and disappear when the mouse leaves the bounding rectangle of the element.
- ☐ Manual control
  Pop-ups begin when the user clicks (presses or releases) the mouse over the element. The RTE determines when to remove the pop-up.

If you select the manual control pop-up behavior, describe the appearance and behavior of pop-ups (text and graphic, as required) in the space below.

This page intentionally left blank

CCT options

Use the following fields to specify the CCT Canvas properties. It is important to note that the CCT options do not offer all of the available runtime features. This is true of many storyboard types, including the Canvas.

Use the spaces below to set the default properties associated with the CCT. Specify options for background size and color, shape, and text style. If needed, support custom styles here.

general/background options

Canvas size (width x height): _____

Background color/opacity %: _____

Autosize background graphic *(y/n)*: _____

general element styles

Default stroke width: _____

Default stroke color: _____

Default text color: _____

Default text font/style/size: _____

Allow custom styles *(y/n)*: _____

action-specific styles

Choose which action types to support in the CCT. The following entries define the color for shapes and text elements, as well as the pixel width of shape elements, for the specified action type.

Jump to storyboard: ☐ _____

Pop-up text: ☐ _____

Pop-up graphic: ☐ _____

Play sound: ☐ _____

Hyperlink: ☐ _____

Command: ☐ _____

"for further exploration"

☐ Command

The Command action type allows the shell to interpret and execute the actions not defined by the Canvas.

If you are using the Command action type, describe the desired actions, including potential parameters, in the space below.

_____

_____

_____

_____

This page intentionally left blank

flash video

The Flash video is a storyboard type that displays both traditional and stepped Flash animations. Flash video has rendered components, but it also typically requires the creation of a "controller" face to allow users to run the animations.

runtime options

Specify those options below. The runtime defines the properties of the Flash video's size and background color.

Size (width x height): _____

Background color: _____

stepped animation support

Using stepped animation requires thought about when to explain the animation content to students. There are two options: play the content and explain what was shown or explain the content and then play. Every step can have associated text; however, there are two special cases when you cannot use text: the initial condition, or the state during which the animation first loads, and the end of the last step. Consider what will occur in those cases.

Choose **one** *of the following options below.*

☐ Steps stop at **"beginning of next step."**
The explanation comes before the animation selection; this means the text displays first.

☐ Steps stop at **"end of current step."**
The explanation comes after the Flash animation; this means the text displays last.

video controller "face"

Although the Flash video has a rendered surface, you must create a player controller to control the Flash animations. Also, consider how you will support primitive controls such as "Play Step," "Play," "Pause," "Fast-forward," "Rewind," and "Pause." Determine the desired functionality and build controls to those specifications. Then combine various controls, such as rewind and play, to create "Go Step 1" and "Play Continuous" functions. Non-stepped, or traditional, animations require a less intricate player.

Describe the video controller's appearance, individual controls, and functionality in the space below.

This page intentionally left blank

“for further exploration”

When loaded, the video player injects references of itself into the Flash video object. This increases the functionality of the Flash video. It can control the video without a controller, embed, use the player to provide control, or use the Flash video player as an additional Flash object to which you can assign text to each Flash component. You can build changes independently, but note that because the Flash object must be embedded with text, changes require redoing and that can slow production.

CCT options

Specify the graphic size (width and height), background color, and use of student text.

Set the properties in the fields below.

☐ Supports stepped animations

Size (width x height): _____

Background color: _____

This page intentionally left blank

flash object

The Flash object storyboard embeds Flash .SWF files designed for one- or two-way communication with the RTE into a storyboard. Defined by the programmers, it pairs reusable Flash animations and simulations with text.

runtime options

If supported, describe the types of configuration data needed and their use in the space below.

If supported, describe what communication will occur between the Flash object and runtime shell in the space below.

CCT options

Specify the Flash communication support and background properties.

Set the properties in the fields below.

☐ Supports initial command

☐ Supports object data

Background graphic size (width x height): _____

Background color/opacity %: _____

This page intentionally left blank

placeholder

A Placeholder is a screen that is common throughout all lessons of a project. Examples of Placeholders include disclaimers and copyright notices. By using a Placeholder, you reduce content development time and minimize potential errors that could occur if you require the content creator to enter the information on a per-lesson basis.

If supporting Placeholders, what types of Placeholders will you need?

Describe the appearance of each supported Placeholder.

CCT options

Determine whether to support Placeholder text by indicating your choice below.

☐ Supports Placeholder text

menu

The Menu can help to organize lessons with top-level organization and multiple pathways. Menus are helpful when employing nonlinear navigation.

runtime options

The Menu storyboard is a face-based player. Specify the appearance of the face in the space below.

How do the storyboard links and screen look? How does the state of the link destination change its appearance? Consider uninitialized, complete, and incomplete states.

CCT options

Specify the upper limit for the number of menu items placed on a storyboard.

Max items: _____

This page intentionally left blank

multiple-choice text

The Multiple-Choice Text storyboard type is a testing storyboard that uses a text question and responses. Because Multiple-Choice Text is a face-based player class, you must establish its appearance; the player itself does not handle the visuals. Content creators can support and format a background graphic and assign remediation to any test storyboard, regardless of testing or question type. See chapter 4 to explore your remediation and scoring options.

runtime options

Specify the appearance of the Multiple-Choice Text face in the space below.

mixed-mode display support

Are you using a mixed mode with some Multiple-Choice Text storyboards having graphics and some not? How does the graphic affect the face appearance? Are storyboards without graphics replaced with another element?

If mixed mode is to be supported, describe how the shell should handle mixed-mode support in the space below.

CCT options

Specify properties for the graphic's size, position, and background in the spaces below.

Support graphic (y/n): _____

Graphic (size and background): _____

This page intentionally left blank

multiple-choice graphics

The Multiple-Choice Graphics storyboard type is a testing storyboard that displays text question and graphic response pairs. Content creators can choose to support a graphic and set the graphic properties. Because the Multiple-Choice Graphics is a face-based player class, you must establish its appearance; the player itself does not handle the visuals. You can also assign remediation to any test storyboard, regardless of testing or question type. See chapter 4 to explore your remediation and scoring options.

runtime options

Specify the appearance of the Multiple-Choice Graphics face in the space below.

CCT options

response element appearance and behavior

Use the spaces below to set the response element behavior and appearance, including size and position.

Graphic size (width x height): _____

Background color: _____

This page intentionally left blank

matching text to text

The Matching Text to Text storyboard type is a face-based testing storyboard that features four to five sets of text-based matching pairs. In addition, there is a "distractor," or incorrect answer. See chapter 4 to explore your remediation and scoring options.

runtime options

face style

Use the space below to specify the Matching Text to Text face properties. Specify options for question and response text size and position as well as size and position of any other elements.

CCT options

Choose whether to support four or five question/answer pairs by indicating your choice below.

☐ Supports fifth question

Graphic (size and background): _____

Background color/opacity %: _____

This page intentionally left blank

matching text to graphics

The Matching Text to Graphics storyboard type is a face-based testing storyboard that features four to five sets of text question/graphic answer matching pairs. In addition, there is a "distractor," or incorrect answer." See chapter 4 to explore your remediation and scoring options.

runtime options

face style

Use the space below to set the face properties. Specify options for question and response text size and position as well as size and position of any other elements.

CCT options

Choose whether to support four or five question/answer pairs and graphics by indicating your choice below.

☐ Supports fifth question

Graphic (size and background): _____

Background color/opacity %: _____

container

The Container storyboard type is actually an aggregation tool, not a storyboard. Although you can view it in the lesson editor like a normal storyboard, it contains no visual or textual content of its own and is not selectable. There are no runtime or CCT properties associated with the Container.

This page intentionally left blank

common storyboard options 4

Common content options span a wide range of functions, including storyboard identification, student instruction, remediation, and navigation. They, like many of Impression's features, are customizable. You can turn off the features that you do not want to support in the CCT using the Project Properties.

general storyboard fields

The following is a list of available general data fields. Specify those you would like to support in the fields below.

☐ Identifier
Optional. Auto-generated, alphanumeric code that uniquely identifies the storyboard.

If supported, determine whether you want the CCT to auto-generate the storyboard identifiers. If so, how? What tokenized string will you use? Note that you do not need to specify the string tokens here, but will need to consider and try to describe what you need.

☐ Objective ID
Identifies the objective associated with the storyboard. Unlike the storyboard identifiers, the CCT does not automatically generate an Objective ID.

☐ Student directions
Directions students see during the lesson like "Click Next to continue."

☐ Alternate directions
A replacement for the student directions when certain criteria are met (for example, when the storyboard is complete).

If employing alternate directions, you will need to consider when they will be used.

☐ Narration file
An audio file associated with the lesson

☐ Narration text
Narration support–either information for a human narrator or modified data for a text-to-speech system

☐ All children complete
If supported, allows content creators to specify that any given storyboard should not be considered complete unless all children (in the lesson map) are marked as complete

☐ User-defined
A field for specifying additional data to be entered with the storyboard

If user-defined is supported, the user should specify the friendly name that the CCT should use for display and the purpose of the field.

This page intentionally left blank

question options

The CCT allows you to mark questions as comprehension checks, ungraded quiz-type questions used for material review, or as test questions, graded questions that determine material comprehension. Determine when you will use each type.

If employing both testing types, what behavioral difference will you associate with the test questions?

response and evaluation handling

Consider how students will provide responses to test questions and/or comprehension checks and how evaluation of those responses will occur. Answer the following questions: How does a student select a response? How does the selected response look? How will the question respond to student selection?

Describe your response and evaluation handling in the space below.

How will the evaluation process work? What will it look like? Define the appearance of incorrect and correct responses. How will students be evaluated? What does your evaluation look like? Can incorrect responses be reset, and if so, how? If students can reset question/answer pairs after answering incorrectly, will all answer pairs be reset or just the incorrect pair?

Describe the evaluation process below.

This page intentionally left blank

remediation options

You can associate remediation with each test question or comprehension check to provide students with feedback on their progress and, if set, review. Impression's remediation options are flexible and offer features like custom messages, storyboard ranges, and combination types. Once set, the particular options provide students with feedback and, in some cases, send students back to the lesson sections for review. You can set remediation in the general tab of the lesson editor.

The following remediation types are available. Specify those you would like to support in the fields below.

standard remediation

☐ None
Uses assigned remediation or default messages

☐ Custom messages
Individually written reply for correct or incorrect responses

☐ Extended custom messages
Individually written responses for correct, partially correct, and incorrect student responses

☐ Matching learning objective storyboards
Uses storyboard learning Objective IDs to link related storyboards together

The CCT itself does nothing; the runtime searches for Objective IDs based on properties set by your runtime builder.

☐ Per-response messages
Individual responses for each correct and incorrect response

☐ Storyboard range
Uses a range of storyboards for review

custom remediation

☐ Combined remediation
Allows the combination of message-type remediation with a storyboard remediation type. The exact combinations depend on the remediation types supported.

This page intentionally left blank

size and appearance

The look and feel of the runtime affects how the created content appears to, and behaves with, the end user. You will need to decide on a number of options, including an optimum screen size for your project, hotspots, the colors of buttons or hotspots, *etc*. Many of these options have been established in chapters 3 and 4.

controlling content

You should consider navigation and mapping types. *Mapping* is the ordering of storyboards within a lesson. Usually, the top-to-bottom display of the storyboards represents the beginning-to-end flow of the lesson. Hierarchical mapping allows you to arrange storyboards into a hierarchical, or parent-child, section-subsection layout. This allows you to enforce a particular order of storyboards. You may prefer to ignore hierarchical organization and use the map simply as an organizational tool.

Linear navigation works like a slide show—following the flow of the lesson as it appears in the map—one storyboard following another. Hierarchical navigation enforces any relationships set up by child-parent relationships established by the hierarchical mapping.

shell commands

In addition to the common content options featured above, you can build and embed custom features into Impression projects. You can also add ancillary capabilities like additional remediation, help, and zoom buttons to your project, which are shell capabilities that can affect the functionality of the runtime. For example, help provides external assistance to the end user, and zoom functions allow users to control the view of the storyboard by magnifying or reducing its size. Other capabilities, such as walk-around, an interactive 360-degree piece, do not affect runtime functionality. To implement ancillary capabilities, decide what CCT options to support and make the necessary changes to the RTE so the can interface.

If supporting additional shell commands, describe those commands and their functionality below.

This page intentionally left blank

runtime deployment options 6

Decisions about content delivery need to be made early. Determining the Learning Management System (LMS) type, communication level, and how content will be packaged and shipped are decisions that you must make with the customer of your courseware. If your customer uses an interface-based LMS like Shareable Content Object Reference Model (SCORM) or Aviation Industry CBT Committee (AICC), your projects must conform to that system's standards. Impression supports SCORM, AICC, and custom LMS interfaces.

LMS interface types and options

Choosing the LMS interface and runtime types and determining how they will communicate are important decisions. The courseware can communicate with the LMS, sending various types of information back and forth between itself and the player. Completion and scoring data, per-answer scores, score aggregation (dividing by objective), lesson storage, and SCORM starting modes are all communication options to consider. In addition to choosing the LMS and runtime, you must also decide what information types need to be sent and stored. Making these decisions early saves time and resources.

There are a number of LMS interface options; typically, the one chosen depends on customer demand. Consult with your customer to determine the best options. For example, do you need SCORM-conformant content? After you decide whether you will need to connect to an interface, consider the types of communication needed and what information needs to be exchanged and returned. Will you need to provide a way to exchange completion and non-completion data? How about bookmarks, test scores, or per-question aggregated scoring? If you are not interfacing with an LMS, you should make decisions about how content will run; for example, using standalone content requires the construction of a courseware launcher. Consult your LMS interface documents for guidance.

What LMS interface type is your project supporting? If custom, describe the predicted functionality and communication types of the LMS in the space below.

deployment types

There are a number of courseware delivery options in addition to the LMS options discussed above. Keeping your LMS type in mind, consider what type of delivery format you will use.

What delivery format(s) should the runtime support? If necessary, also describe the launching mechanism.

This page intentionally left blank

media organization

Media organization is the arrangement of media assets in your project. You can choose to place all of the media files in a single, central repository, referred to as the media path. You can also create subdirectories, dividing media assets into smaller, more specific storage areas. The CCT supports subdirectories for media paths that "match" the hierarchical layout of the course content.

*Choose **one** of the following media organization options.*

☐ Supports flat media path ☐ Supports subdirectories

If supporting a flat media organization, specify its root path below.

_____

If supporting subdirectories, describe their organization in the space below.

_____

_____

_____

_____

course outline flow

The course outline is the organization of lessons within the CCT. A lesson is an individual unit of training content consisting of zero or more storyboards that you should edit using the CCT's lesson editor. A group is an aggregation of zero or more lessons or groups, provided solely as an organizational tool. You cannot export groups, nor do they contain data other than a basic title and an identifier code.

You can choose to support the hierarchical organization, but you must decide how to implement and enforce it. Remember to organize your lessons logically; the more lessons you have, the more groups you should create. This way, your content stays neat and organized.

Use the space below to specify your project's course outline.

_____

_____

_____

_____

This page intentionally left blank