content creation tool
administrator's guide

May 2008

impression

table of contents

introduction

The Impression Content Creation Tool Administrator's Guide is a "behind the scenes look" at the Impression Content Creation Tool (CCT) and how it can be customized to fit your project's specific needs.

You will learn about the database types available for use; how to create, delete, and manage those databases; and assign rights to users via the Impression Connector Management Tool.

You will set production properties and support preview using the Project Properties dialog, and create and manage lesson content using the Lesson/Group Editor.

Limiting the available features, fields, and media types will ensure that your content creators will see and use only those that are appropriate to your project. All of this will culminate into production simplification and time-savings.

Chapter 1
impression databases

Impression database architecture

You will need a basic understanding of Impression's database system to select the database best for your project. Impression uses both Microsoft Access and Microsoft SQL Server databases.

What is an Impression database? An Impression database is a repository where lesson content created with the Impression Content Creation Tool (CCT) is stored. The CCT can use content databases created in either Microsoft Access or SQL Server. Content databases should be created on a per-project basis since customization options apply to all content stored within the database.

Using the information in this chapter, you will be able to create and manage Microsoft Access and SQL Server content databases, and the Impression connector databases, plus establish user rights.

Access rights in Microsoft Access and SQL Server

Access rights are defined on a per-user basis for each content database.

The CCT supports the concept of access rights. Access rights determine the tasks users are allowed to perform; for example, limited access rights reduce the number of content databases available for a user to open and edit. Microsoft Access-based .MDB databases do not support per-user rights, but SQL Server databases do. The Impression connector database enforces the access rights for SQL Server content databases.

The following SQL Server database access rights are defined:

- *Content Creators/Users* edit content in databases where they have been assigned rights.
- *Administrators* can perform all actions that users can perform. In addition, Administrators can also add, remove, and rearrange lessons and groups, change

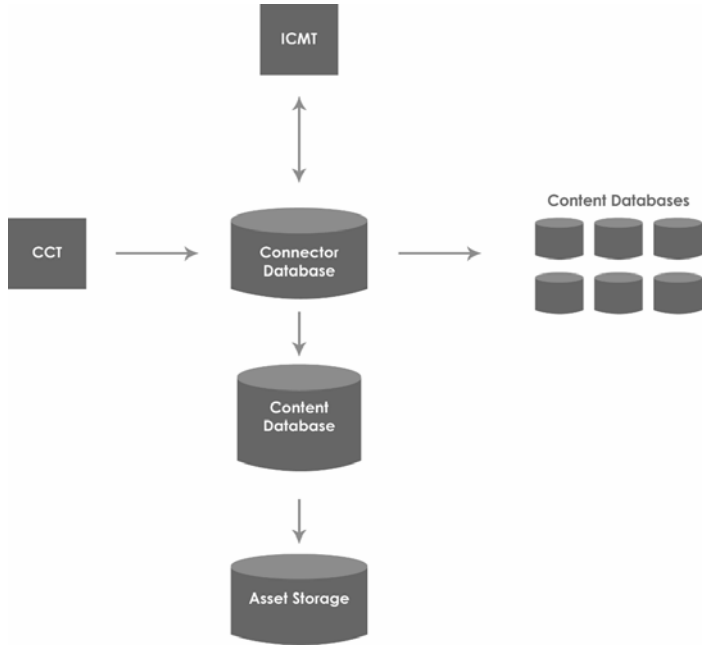
Project Properties, edit content connection information, add new users, and add and remove access rights (including administrative access) for users. .

- *Superusers* can act as administrators for all content databases defined on a server. They can create superusers, users, and administrators, permanently remove users from the system, and create and delete content databases and database connection information.

How does Impression work with SQL Server?

Impression uses the SQL Server login created by the IT staff to access the connector database. The account information includes a user ID and password that is stored in the computer registry as plain text. This account needs, at the very least, read-only access to the Impression tables in the connector database. If the IT department is going to allow administrators and superusers to access the Impression Connector Management Tool (ICMT), the account needs to have read/write access to the tables in the connector database.

SQL Server Database Structure



Impression connector database

The connector database provides the CCT with information about the content databases stored on the SQL Server and identifies which databases the currently logged-in user has access to, as well as their access rights for each database. Your IT personnel will provide you

with the connection information you need to access the connector. This information includes the machine name as well as the login ID and password of the SQL Server account authorized to access the connector.

Creating an Impression connector database

To create the impression connector database, run the script file **`Utilities\impressionconnector SQL Server Install Script.sql`** in Microsoft SQL Query Analyzer. This script, located on the disk image, needs to be run by a user with SQL Server System Administrator rights. The script sets a default login and password of 'impression' and 'impression' respectively. This login and password combination is used by the CCT to connect to the connector. The **`Utilities\impressionconnector SQL Server Install Script.sql`** script creates the following: a) a username and password, b) a connector database, impressionconnector, c) a stored procedure for creating new databases, used by the ICMT, and d) a stored procedure for creating new content databases from the SQL Server console.

How to create an Impression connector database on SQL Server:

- Open the script, **`Utilities\impressionconnector SQL Server Install Script.sql`** in SQL Query Analyzer.
- Read through the script to become familiar with what it does.
- To modify the default login and password used to connect to the Impression connector database, follow the instructions that are marked by the delimiter

  `-- USER ACTION --`
- Run the script. The Query Analyzer Messages tab will provide feedback about the script's completed action and the SQL Server Login/Password that will be used in the Impression CCT to connect to SQL Server.

SQL Query Analyzer connector database confirmation

```
The CREATE DATABASE process is allocating 0.63 MB on disk 'impressionConnector'.
The CREATE DATABASE process is allocating 0.49 MB on disk 'impressionConnector_log'.


New login created.

Granted database access to 'Impression'.


******************************************************************************
The Impression database, impressionConnector, has been created in SQL Server.
Please setup a maintenance plan to safeguard your impression data.

The login and password to use for the SQL Server setup in the Impression
application is 'Impression', 'Impression', respectively.
******************************************************************************
```

SQL Server content databases

An SQL Server content database is an Impression content database served by a machine running Microsoft SQL Server. Its structure is identical to an Impression content database stored in Microsoft Access .MDB format. Unlike an Access-formatted content database, SQL Server content databases are accessed indirectly through the Impression connector database, which also manages user access to the SQL Server content database.

A new database should be created for each project. Once the connector database has been created on SQL Server, this can be done by calling the **`NewImpressionDBFromConsole`** stored procedure connector database through SQL Query Analyzer.

How to create an Impression content database using SQL Server Query Analyzer:

- Use SQL Server Query Analyzer to connect to the `impressionconnector` database. Note that the login will require system administrator privileges to create the project database using the stored procedure.
- Call **`the proc_NewImpressionDBFromConsole`** with the name for the project database, SQL Server login and password.

```
EXEC proc_NewImpressionDBFromConsole 'ProjectName', 'SQLLoginName',
'SQLLoginnamePassword'
```

- The Query Analyzer Messages tab will provide feedback about the creation of the new Impression content database. If this database does not exist in the connector, you will need this information in the ICMT to make the database available to users.

SQL Server Query Analyzer content database confirmation

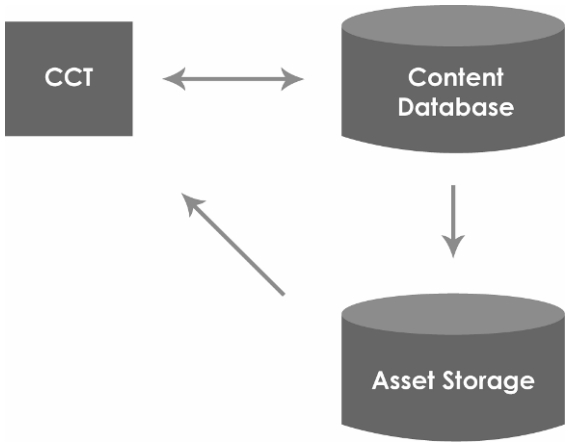
```
The CREATE DATABASE process is allocating 0.63 MB on disk 'NEWIMPRESSIONCONENTDATABASE'.
The CREATE DATABASE process is allocating 0.49 MB on disk 'NEWIMPRESSIONCONENTDATABASE_log'.

Granted database access to 'Impression'.


The Impression database, NEWIMPRESSIONCONENTDATABASE, has been created in SQL Server.
Please setup a maintenance plan to safeguard your impression data.
```

Microsoft Access .MDB content databases

An .MDB content database is an Impression content database stored in Microsoft Access-compatible .MDB format. User access rights are not supported for Access content databases;

anyone who can open the file is automatically considered an administrator and has full control over the data. Note that Microsoft Access is not required to create and use .MDB content databases.

Microsoft Access Database Structure



How to create an .MDB content database:

- From the File menu of the Groups Window, select *New Database*. The Create New Impression Database dialog will be displayed.
- Use the dialog to specify a directory (folder) and filename for the content database, then choose *Save*. A new .MDB content database with the name you chose will be created at the location you specified. Once created, the CCT will automatically open the new database.

Tip: If the *New Database* menu item is not visible, the CCT is currently in SQL Server mode. Use the *Change Database Mode* menu item available from the File menu to switch to .MDB (Microsoft Access) mode.

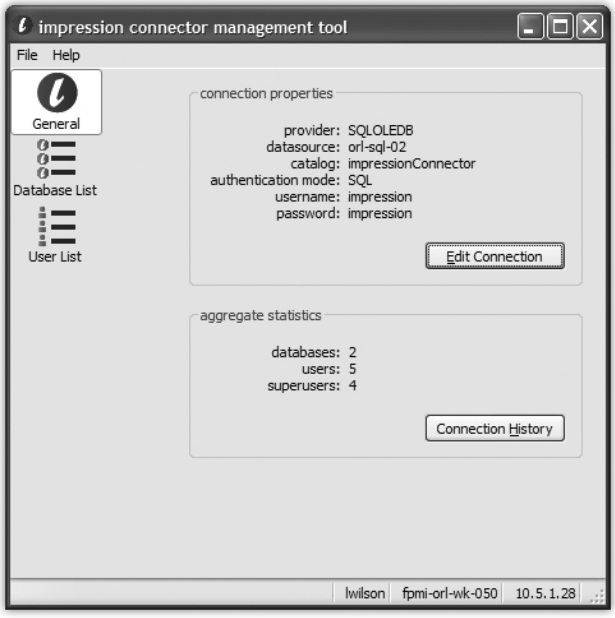
Chapter 2
impression connector management tool

Overview

The Impression Connector Management Tool (ICMT) manages the information stored in the connector database. Databases and users can be added, managed, and deleted in the ICMT. You will only need to use the ICMT while working in SQL Server mode. Access the ICMT by selecting *Connector Management* from the Groups window Tools menu.

As discussed earlier, there are three categories of users: user, administrator, and superuser. Users have the most limited access; administrators have some privileges, such as adding and deleting users within their database; and superusers have unlimited access to user and database management for all SQL Server databases.

Impression Connector Management Tool



ICMT Database Management

You can use the ICMT to create new databases or database references, change reference information, or remove database references.

How to create an SQL Server content database using the ICMT:

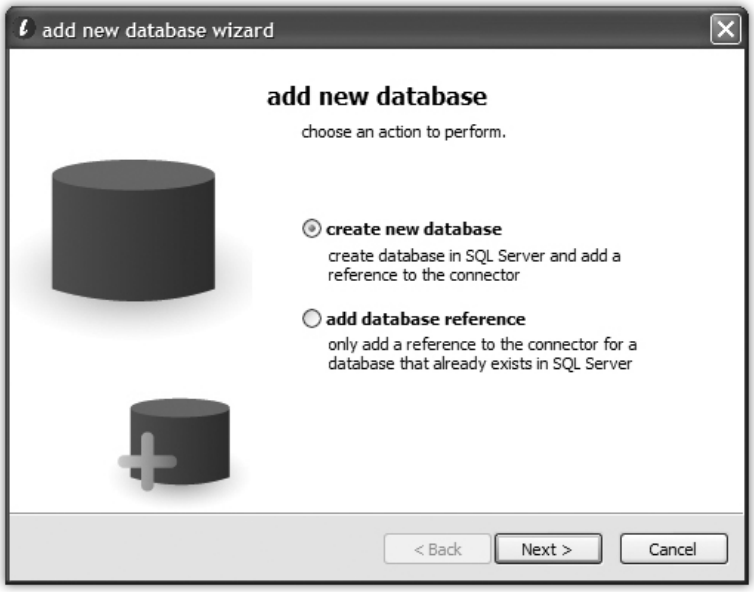
- Select the *Database List* button from left menu and then *Add…* to start the add new database wizard. Choose the *create new database* option and then select *Next*.
- Fill in the information boxes for catalog name, username, password, title, and description. All of the fields, except the description are required. Select *Next*.
- The following checks occur during the database settings verification: a) catalog name does not already exist in the connector; b) the database does not already exist in SQL Server. If the verification passes, you will be prompted to enter an SQL Server login that has `create database` and `grant` privileges. Select *Next*.
- If there are no errors, the database is created in SQL Server and a reference is added to the connector database. Click *Close* to continue.

How to add an SQL Server content database reference:

- Select the *Database List* button from the left menu and then *Add…* to start the add new database wizard. Choose the *add database reference* option and then select *Next*.
- Fill in the information boxes for catalog name, username, password, title, and description. All of the fields, except the description are required. Select *Next*.
- The following checks occur during the database settings verification: a) catalog name does not already exist in the connector; b) the database already exists in SQL Server.
- If there are no errors, the database reference is added to the connector database. Click *Close* to continue.

Tip: Be sure to notify users of database edits, particularly titles, as those are selected by the users where databases are selected for editing.

Add new database wizard



How to edit an SQL Server content database reference:

- Select the *Database List* button from the left menu and then select the database you want to edit from the catalogs list. Fill in the new information for catalog name, username, password, title, and description. All of the fields, except the description are required.
- Click on the *Test Connection* button. Once the database passes the validation process, the Save button is enabled. Select the *Save* button. Be sure to notify users of database edits, particularly titles, as those are selected by the users where databases are selected for editing.

How to remove an SQL Server content database reference:

- Select the *Database List* button from left menu and then *Delete*. The confirm database delete dialog will appear. To continue the deletion process, type `confirm` into the dialog. Finally, select *OK*. The dialog will disappear and the reference to the database will be removed from the connector database

Note: Removing an SQL Server content database does not delete it from the SQL Server; it simply removes the reference to the database from the connector. If you would like to permanently delete the content database, you must delete the database using SQL Server's tools.

ICMT User Management

Use the ICMT to add, delete, and edit user access rights to databases. Access rights can be defined on a per-user basis for each database. This means that a content creator can have administrative rights to one content database and user rights to another. The SQL Server connector database enforces the access rights for SQL Server content databases.

Adding users to the connector database using the ICMT

As an administrator or superuser, you can use the ICMT to add users to the connector and content databases. Administrators can grant or deny access rights to both new and existing users for those databases where they have administrative rights. Superusers can grant or deny superuser rights as well as administrative or user rights. Superusers can permanently remove a user's information from the connector database.

Access rights determine what tasks users are allowed to perform; for example, limited access rights reduce the number of content databases available for a user to open and edit.

Tip: For the user being edited, rights changes do not take immediate effect—they will take effect the next time the user runs the CCT or ICMT.

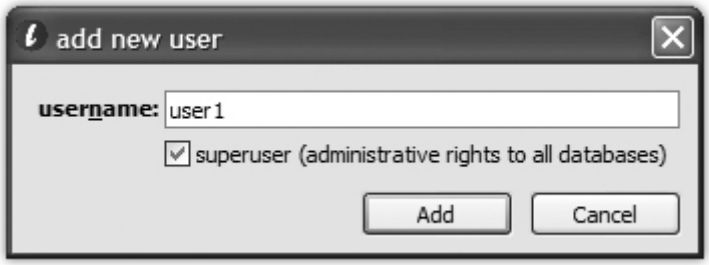
To add a superuser:

- Select the *User List* button from the left menu and select the *Add...* button. Fill in the username field with the user's Windows login ID. Select the *superuser (administrative rights to all databases)* option to grant superuser rights and click *Add*.

To add a user:

- Select the *User List* button from left menu and select the *Add...* button. Fill in the username with the Windows logon name of the user. Uncheck *the superuser (administrative rights to all databases)* and then click *Add*. This action will add a user but will not grant them rights to any databases—grant access by following the procedures below.

Use the add new user dialog to assign user or superuser status to content creators.



Assigning access rights to a content database

You can assign access rights from either the User List tab or the Database List tab.

To add administrative access rights to a database for a user:

- Select the *User List* button from the left menu. Highlight a username from the available list and select *Catalogs*.
- When the catalogs for [user] dialog displays, assign administrator access rights to a database by selecting the *administrator access rights* option next to each database that you want to grant administrator rights to and then select *Save*.

  —Or—
- Select the *Database List* button from the left menu. Highlight a database and select *Users*.
- When the users for [database] appears, select the administrator option from the access type drop-down menu. Click the box next to each user name that you want to grant administrator rights to and then select *Save*..

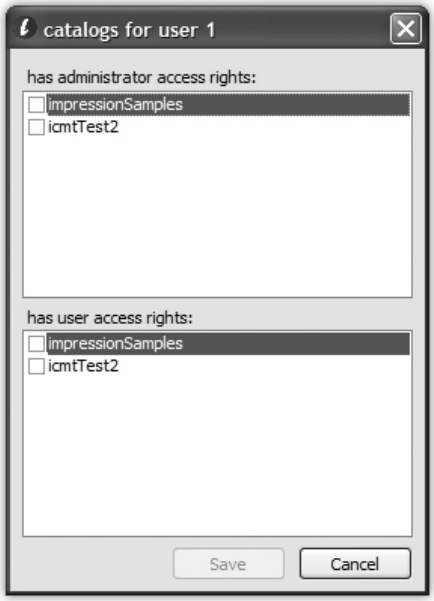
To add content creator rights to a database for a user:

- Select the *User List* button from the left menu. Highlight a username from the available list and select *Catalogs*.
- When the catalogs for [user] dialog displays, assign user access rights to a database by selecting the *user access rights* option next to each database that you want to grant user rights to and then select *Save*.

  —Or—
- Select the *Database List* button from the left menu. Highlight a database and select *Users*.

- When the users for [database] dialog appears, select the *user* option from the access type drop-down menu. Click the box next to each username that you want to assign *user access* to and then select *Save*.

Use the catalogs for [user] *dialog to edit user rights.*



Editing user access rights

How to change a user to a superuser:

- Highlight the username from the available list, and then select *Edit*. When the edit user information dialog appears, click the *superuser* box and select *Save*.

Tip: Only a superuser can promote a user to a superuser. Administrators do not have the access rights to create superusers.

How to change a user to an administrator:

- Follow the procedure for adding a user to the database; only change user rights to administrator rights by checking the boxes for *administrator access rights*.

Tip: Administrators can change a user to an administrator in a database where they are administrators; however, superusers can change any user's access rights in any database.

Removing user access

Important Notes:

1. Only a superuser can delete another superuser; even an administrator within the database the superuser is being removed from does not have this right.
2. Removing superuser status removes the user's access rights to all databases. Be sure to reassign the appropriate user status.

To remove content creator rights for a user:

- Select the *Database List* button from the left menu and then select a database. After the database is selected, select *Users*. When the users for [database] dialog appears, select the *user* option from the access type drop-down menu. Remove the checkmark next to the username and then select *Save*.

To remove administrative rights for a user:

- Select the *Database List* button from the left menu and then select a database. After the database is selected, select *Users*. When the users for [database] dialog appears, select the *administrator* option from the access-type drop-down menu. Remove the checkmark next to the username and then select *Save*.

How to remove user and administrator rights using the Users List:

- Select the *Users List* button from the left menu and then select a user. After the user is selected, select *Catalogs*. When the catalogs for [user] dialog appears, remove the checkmark next to each database from where you want to remove user/administrator access and then select *Save*.

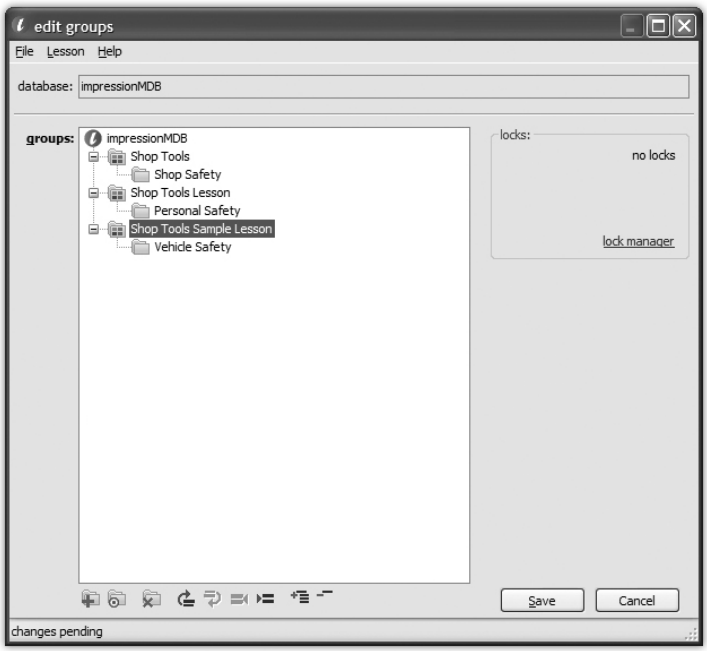
Tip: Database administrators can change a user to an administrator in a database where they are administrators; however, superusers can change any user's access rights in any database.

Chapter 3
lesson and group management

Overview

Using the Lesson/Group Editor located in the Groups Window, administrators can add, delete, and manage groups and lessons. Note that changes made in the Editor are not immediate; they take effect when you select the *Save Changes* button.

Lesson/Group Editor



Lessons and Groups

A **lesson** is an individual unit of training content. Lessons consist of zero or more storyboards and are edited using the CCT's Lesson Editor. If supported, a lesson may also contain a companion Lesson Guide, a single page of HTML content used for instructors.

A group is an aggregation of zero or more lessons or groups and is provided solely as an organizational tool. Groups are not exportable, nor do they contain data other than a basic title and user-defined identifier code.

Lesson/Group editor

The Lesson/Group Editor is a tool for managing lessons and groups. You can add, edit, and delete both lessons and groups using the editor. You can access the Lesson/Group Editor from the Tools menu in the Groups Window. When you select the Lesson/Group Editor menu item, you are prompted to confirm that you want to make changes. Do this by typing *confirm* into the dialog box and then choosing *OK*.

Note that if you make destructive changes while editing—for example, deleting a lesson with content—you will again be prompted to confirm the changes before your actions are committed to the database.

How to add lessons/groups:

- From the Lesson/Group Editor, choose *Add New Lesson/Group...* from the Lesson menu. When the add/edit group dialog appears, enter the title of the lesson or group into the textbox provided. If you wish, enter an identification code for the lesson or group into the identifier textbox.
- If this item is to be a lesson, ensure that the *group is a lesson (supports storyboards)* checkbox is checked. If this item should be a group, clear the checkbox.
- Choose *OK* to add the lesson or group or *Cancel* to dismiss the dialog without adding the item.

Managing lessons/groups

The Lesson/Group Editor can also be used to establish lesson and group organization. Editing commands, including hierarchical organization commands, are available from the Lesson menu.

How to manage the lesson/group hierarchy:

Use the items on the Lesson menu of the Lesson/Group Editor to manage the hierarchy. The following movement options are available:

- Up
- Down
- Indent
- Outdent

Removing lessons/groups

You can delete lessons/groups using the Lesson/Group Editor's Lesson menu or editing icons.

How to remove lessons and groups:

- To remove a lesson or group, highlight the lesson that you would like to delete and select *Delete* from the Lesson menu. When the delete lesson/group dialog appears, select *Yes*.

How to remove lessons/groups with associated lessons/groups

- To remove a lesson or group, highlight the lesson that you would like to delete and select *Delete* from the Lesson menu. When the delete multiple lessons/groups dialog appears, select *delete selected lesson and children* or *delete selected lesson only*.
- To remove a lesson or group, highlight the lesson that you would like to delete and select *Delete* from the Lesson menu. When the delete multiple lessons/groups dialog appears, select *delete selected group and children* or *delete selected group only*.

Tip: If deleting an item that has children, or indented storyboards, associated with it, you will be prompted to confirm your deletion. If you delete the parent, but not the children, by selecting *delete selected item only*, the children will be promoted to the original parent level.

Chapter 4
project properties

Overview

The Project Properties editor, accessible from the *Tools* menu of the Groups window, is used to format and customize the Content Creation Tool for a specific project. You can use the Project Properties editor to define media asset locations, report locations, export manifest templates, and lesson guide templates. Only users with administrative rights can edit Project Properties.

Select *Project Properties* from the Tools menu of the Groups window to access and edit your project properties.

Navigation and selection

Use the tab menu on the left of the Project Properties Editor to access the option categories.

Saving changes

Remember that changes made in Project Properties do not take immediate effect. Save your changes by selecting the *Save Changes* button.

Cancelling changes

If you change your mind about a feature modification, cancel changes by selecting the *Cancel* button. However, you should note that once changes have been applied via the *Save Changes* button, they cannot be cancelled.

Resetting defaults

To restore the default settings, select *Reset Property Values* via the File menu in Project Properties.

Importing/exporting

Import preexisting CCT project properties data or export the current project properties by using the commands on the File menu.

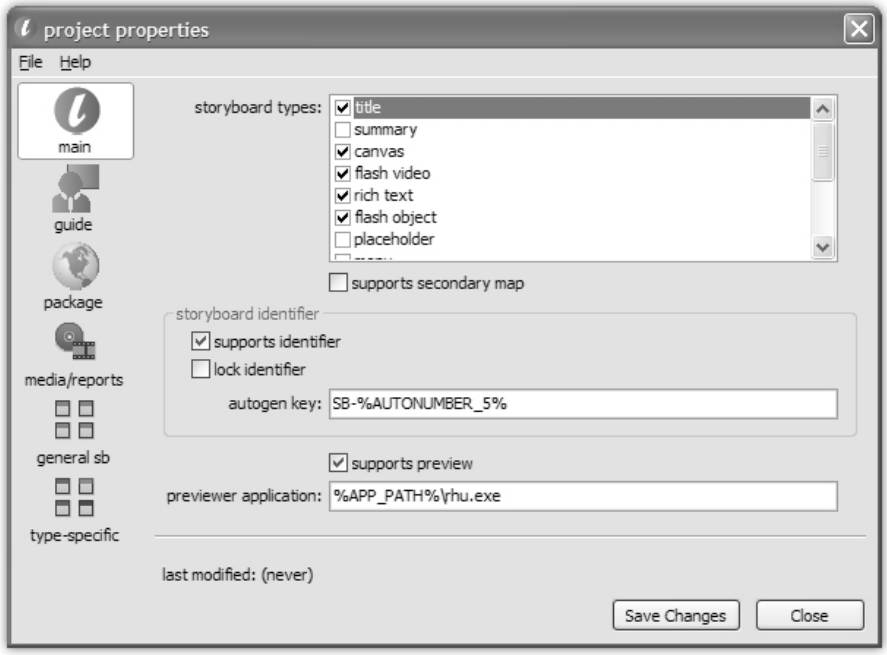
Note: Project Properties is so named because it is project-specific. Due to the large number of project property options, administrators should consult *The Impression New Project Manager's Workbook* to decide which features to enable and what values to set.

Using Project Properties makes content creation and editing easier by allowing you to turn off the features that you do not need to use for a specific project. Turning off unwanted fields cuts down on confusion because your users will only see the options fields appropriate to your specific project.

Main properties

In the Main category, you can choose storyboard types and functions, support for a secondary map, define storyboard identifier features, and associate a previewer with the project.

Main properties



Main property options

The following items are found in the Main tab of the Project Properties dialog:

Storyboard Types

Choose the storyboard types for your project by placing a checkbox next to each type you wish to support.

Supports Secondary Map

Check this item to add a second storyboard map to each lesson.

Supports Identifier

Check this box if you intend to use an alphanumeric code to identify each storyboard in a lesson.

Lock Identifier

If checked, prevents the identifiers from being edited. Use this in conjunction with the *Autogen Key* field.

Autogen Key

The *Autogen Key* is a tokenized string used to generate identifiers for new storyboards. This item accepts the following tokens:

Token	Description
`%LESSON_TITLE%`	The title of the current lesson.
`%LESSON_IDENTIFIER%`	The identifier of the current lesson.
`%NEW_GUID%`	A new Globally Unique Identifier (GUID) of the form "xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx" that is a 128-bit pseudo-random number generated using an algorithm to minimize the possibility of duplication.
`%AUTONUMBER%`	A long integer (32-bit whole number) retrieved from the current content database's **`impressionAutonumbers`** table. This number should be unique enough to differentiate between storyboards.
`%AUTONUMBER_1%` through `%AUTONUMBER_9%`	The same value as %AUTONUMBER%, but formatted to ensure a minimum length based on the value of the trailing digit. For example, if the next available autonumber is 17, %AUTONUMBER_5% would generate the string "00017".
`%SBID%`	The GUID associated with the storyboard.

Supports Preview

If checked, allows one-click preview of lessons or storyboards from both the Lesson Editor and the Groups Window.

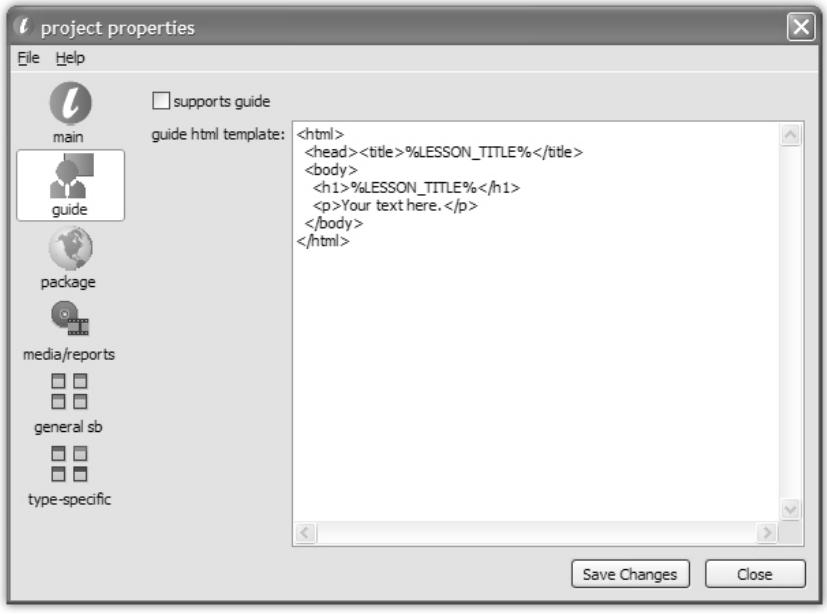
Previewer Application

Specify the path and filename of the previewer application to use for your project. Note that the preview command line only specifies the actual executable to use for the project's previewer. For more information, refer to Chapter 6.

Guide properties

The Guide category contains Lesson Guide-related properties. A Lesson Guide is an HTML document intended to serve as a companion to an individual lesson. Guides can be synchronized with a runtime engine (the Impression RTE DevKit contains an example).

Guide properties



Guide property options

The following items are found in the Guide tab of the Project Properties dialog:

Supports Guide

Check this box if your project plans to use Lesson Guides.

HTML Template

Use this textbox to enter the default HTML text that should be used when a guide is first edited. Note that this data is *only* used the first time that a guide is accessed. If a guide has

already been edited, changes made to the template will not appear in later versions of the guide.

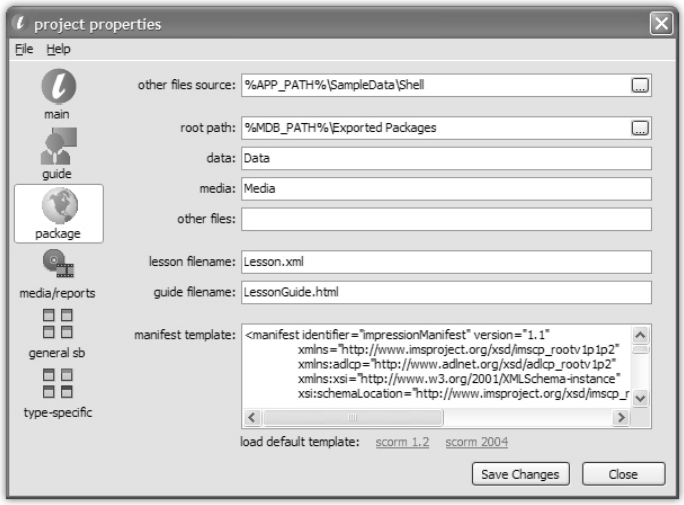
The HTML Template field supports the use of tokenized data. The following tokens will be recognized and replaced the first time that the guide is edited:

Token	Description
%APP_PATH%	The location of the CCT.
%.MDB_PATH%	The location of the database in .MDB mode. In SQL Server mode, this token is replaced with the empty string.
%LESSON_TITLE%	The title of the current lesson.
%LESSON_IDENTIFIER%	The identifier of the current lesson.

Package properties

Packaging is the process of preparing lesson content, including lesson and guide data, media assets, and runtime files for deployment. Use the Package category to set options such as the name and location of the output root path and to specify the location for data, media, and other file assets. A manifest template can also be specified.

Package properties



Package property options

The following items are found in the Package tab of the Project Properties dialog:

Other files source

A tokenized string that identifies the location of additional assets to be copied to the output package directory, such as runtime files. The *other files source* field accepts the following tokens:

Token	Description
%APP_PATH%	The location of the CCT.
%MDB_PATH%	The location of the database in MDB (Access database) mode. In SQL Server mode, this token is replaced with the empty string.

Package root path

A tokenized string that identifies the main package output directory. The *package root path* field accepts the following tokens:

Token	Description
%LESSON_TITLE%	The title of the current lesson. Invalid filename characters (including any backslash characters) present in the title are replaced by the underscore ("_") character.
%LESSON_IDENTIFIER%	The identifier of the current lesson. Invalid filename characters (including any backslash characters) present in the identifier are replaced by the underscore ("_") character.
%PARENT_TITLE%	The title of the parent group or lesson as shown in the Groups Window. Invalid filename characters (including any backslash characters) present in the title are replaced by the underscore ("_") character.
%PARENT_IDENTIFIER%	The identifier of the parent group or lesson as shown in the Groups Window. Invalid filename characters (including any backslash characters) present in the identifier are replaced by the underscore ("_") character.
%LESSON_TITLE_PATH%	A backslash-delimited list of lesson/group titles from the current lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual title are replaced by the underscore ("_") character.

Token	Description
%LESSON_IDENTIFIER_PATH%	A backslash-delimited list of lesson/group identifiers from the current lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual identifier are replaced by the underscore ("_") character.
%PARENT_TITLE_PATH%	A backslash-delimited list of lesson/group titles from the parent group or lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual title are replaced by the underscore ("_") character.
%PARENT_IDENTIFIER_PATH%	A backslash-delimited list of lesson/group identifiers from the parent group or lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual identifier are replaced by the underscore ("_") character.
%APP_PATH%	The location of the CCT.
%MDB_PATH%	The location of the database in .MDB mode. In SQL Server mode, this token is replaced with an empty string.
%DATABASE_TITLE%	In .MDB mode, the name of the .MDB file (without extension). In SQL Server mode, the friendly title of the catalog. Invalid filename characters (including any backslash characters) present in the title are replaced by the underscore ("_") character.

Packager subdirectories

Each class of file associated with the packaging process can be placed into its own subdirectory. The data, media, and other files stored in the database or on a file server in their respective directories are loaded into the packager root path during packaging. The *data* and *media* fields identify the subdirectories, relative to the output root path, where these files will be stored. These fields are not tokenized.

The *lesson filename* and *guide filename* fields allow you to tokenize filenames for the Lesson and Guide .

These fields accept the following tokens:

Token	Description
%LESSON_TITLE_PATH_FILENAME%	An underscore-delimited list of lesson/group titles from the lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual title are replaced by the underscore ("_") character.
%LESSON_IDENTIFIER_PATH_FILENAME%	A backslash-delimited list of lesson/group identifiers from the lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual identifier are replaced by the underscore ("_") character.
%PARENT_TITLE_PATH _FILENAME%	An underscore-delimited list of lesson/group titles from the parent group or lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual title are replaced by the underscore ("_") character.
%PARENT_IDENTIFIER_PATH_FILENAME%	An underscore-delimited list of lesson/groups identifiers from the parent group or lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual identifier are replaced by the underscore ("_") character.

SCORM manifest template

You can also choose to generate a manifest when a lesson is packaged. The CCT includes simple default manifest templates for SCORM 1.2 or SCORM 2004. You can use these templates "as-is," or create your own. Click on the *SCORM 1.2* or *SCORM 2004* links to fill in the manifest template field with a default template.

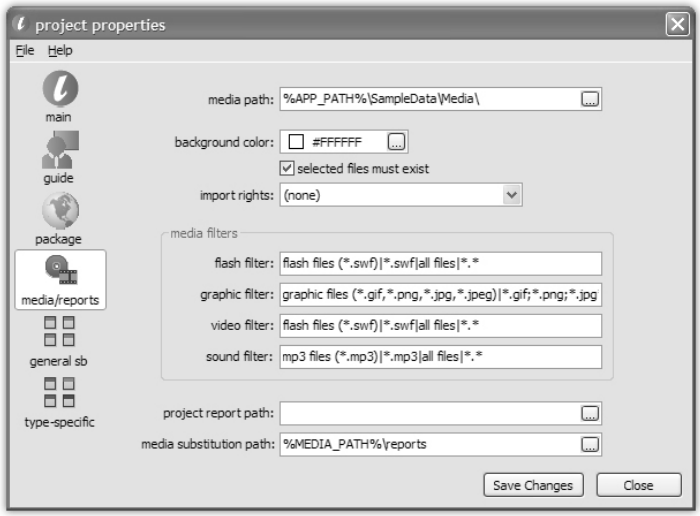
The *SCORM manifest template* field accepts the following tokens:

Token	Description
%LESSON_TITLE%	The title of the current lesson. Invalid filename characters (including any backslash characters) present in the title are replaced by the underscore ("_") character.
%LESSON_IDENTIFIER%	The identifier of the current lesson. Invalid filename characters (including any backslash characters) present in the identifier are replaced by the underscore ("_") character.
%PARENT_TITLE%	The title of the parent of the current lesson. Invalid filename characters (including any backslash characters) present in the title are replaced by the underscore ("_") character.
%PARENT_IDENTIFIER%	The identifier of the parent of the current lesson. Invalid filename characters (including any backslash characters) present in the identifier are replaced by the underscore ("_") character.
%LESSON_DATA_LIST%	The manifest XML-formatted string listing all lesson data files.
%LESSON_MEDIA_LIST%	The manifest XML-formatted string listing all media asset files.
%OTHER_FILES_LIST%	The manifest XML-formatted string listing the contents of the "other files" folder.

Media/Reports properties

The Media/Reports category contains options for selecting and organizing media and reports. Here, you can choose the types of media that will be available for selection in your lessons, the location of the media files, and the location of additional reports.

Media properties



Media property options

The following items are found in the Media tab of the Project Properties dialog:

Media path

A tokenized string that identifies the location of the media assets that can be selected for the project. You can choose to have all lessons use the same file system directory, or you can define the media path so that each lesson has its own directory.

Use the ellipsis button on the right side of the media path textbox to choose an existing path.

The *media path* field accepts the following tokens:

Token	Description
`%LESSON_TITLE%`	The title of the current lesson. Invalid filename characters (including any backslash characters) present in the title are replaced by the underscore ("_") character.
`%LESSON_IDENTIFIER%`	The identifier of the current lesson. Invalid filename characters (including any backslash characters) present in the identifier are replaced by the underscore ("_") character.

Token	Description
%PARENT_TITLE%	The title of the parent group or lesson as shown in the Groups Window. Invalid filename characters (including any backslash characters) present in the title are replaced by the underscore ("_") character.
%PARENT_IDENTIFIER%	The identifier of the parent group or lesson as shown in the Groups Window. Invalid filename characters (including any backslash characters) present in the identifier are replaced by the underscore ("_") character.
%LESSON_TITLE_PATH%	A backslash-delimited list of lesson/group titles from the current lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual title are replaced by the underscore ("_") character.
%LESSON_IDENTIFIER_PATH%	A backslash-delimited list of lesson/group identifiers from the current lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual identifier are replaced by the underscore ("_") character.
%PARENT_TITLE_PATH%	A backslash-delimited list of lesson/group titles from the parent group or lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual title are replaced by the underscore ("_") character.
%PARENT_IDENTIFIER_PATH%	A backslash-delimited list of lesson/group identifiers from the parent group or lesson up to the database root. Invalid filename characters (including any backslash characters) present in any individual identifier are replaced by the underscore ("_") character.
%APP_PATH%	The location of the CCT.
%MDB_PATH%	The location of the database, in .MDB mode. In SQL Server mode, this token is replaced with the empty string.

Token	Description
%DATABASE_TITLE%	In .MDB mode, the name of the .MDB files (without extension). In SQL Server mode, the friendly name of the catalog. Invalid filename characters (including any backslash characters) present in the title are replaced by the underscore ("_") character.
%DATABASE_CATALOG%	In .MDB mode, the name of the .MDB files (without extension). In SQL Server mode, the actual database catalog name. Invalid filename characters (including any backslash characters) present in the catalog name are replaced by the underscore ("_") character.

Background color

The Background color property allows you to specify the background color of the Asset Picker window. This is useful if you're using files (like .PNG or .GIF) that support transparency.

Selected files must exist

If this item is checked, users can only select items from the Asset Picker's list of files (or an empty filename). If this item is unchecked, users can type in any value for a filename.

Import rights

Use the Import rights dropdown list to specify which types of users are allowed to import graphics into the media folder from the Asset Picker. You can choose from all users, administrators only, or no users.

Note: If you enable import rights, make sure that the users who are given these rights have the appropriate filesystem permissions. If you don't know if the users have filesystem permissions, check with your system administrator.

Media asset filter strings

The filter string entries allow you to specify the types of files that are displayed in the Asset Picker for each different type of asset. The combo box at the bottom left of the Asset Picker can be used to switch between filter values, just like the standard Windows file selector dialog.
The filter string format is:

```
description"|"filter ["|"description"|"filter]
```

This string is used when description is a textual description of the filter directly following the pipe ("|") character and filter is the wildcard-based selection criteria.

Examples:

If you want to display all .GIF files, you can use the following filter string:

```
GIF Files|*.gif
```

To display two filter strings, one for .GIF files and one for .PNG files, you can use the following filter string:

```
GIF Files|*.gif|PNG Files|*.png
```

To display two different sets of files in one filter string, separate the wildcard-based selection criteria with a semicolon (";"):

```
GIFs and PNGs|*.gif;*.png
```

To display all files, use "*.*" for the filter parameter:

```
All Files|*.*
```

Project report path

A tokenized string that identifies the location of project-specific reports. Reports found in this directory are added to the Report Wizard along with any reports found in the CCT installation directory.

Use the ellipsis button on the right side of the project report path textbox to choose an existing path.

The *project report path* field accepts the following tokens:

Token	Description
%MEDIA_PATH%	The media path of the current lesson.
%APP_PATH%	The location of the CCT.
%MDB_PATH%	The location of the database, in .MDB mode. In SQL Server mode, this token is replaced with the empty string.

Media substitution path

A tokenized string that identifies the alternate location of media assets to be used for reports. Whenever a media element (graphic file, video file) is needed for a report, this location is checked to see if the file exists here. If it does, the file in this location is used. If the file is not found, the file in the normal media path location is used.

Use the ellipsis button on the right side of the media substitution path textbox to choose an existing path.

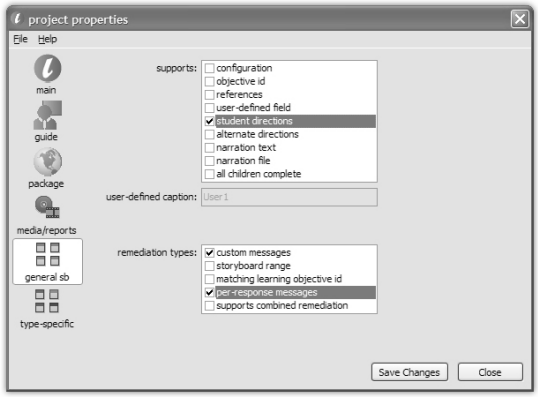
In addition to accepting all tokens that the *media path* field accepts, the *media substitution path* field accepts the following tokens:

Token	Description
%MEDIA_PATH%	The media path of the current lesson.

General Storyboard properties

The General Storyboard category contains options that affect the fields associated with common storyboard options shared across storyboard types. These fields are unlike the type-specific options, which only affect the given storyboard type.

General Storyboard properties



General Storyboard property options

The following General (storyboard) tab fields can be enabled or disabled. Check or uncheck each of the following items to show or hide them in the Lesson Editor's General tab:

Configuration

The Configuration field is a user-defined field for creating additional CCT capability. For example, a link to an external information source or additional shell formatting can be stored in Configuration.

Objective ID

The Objective ID is an alphanumeric code that identifies the learning objective associated with a storyboard. It is useful for "Matching Learning Objective Storyboards" remediation, or for reporting detailed scoring data to a Learning Management System (LMS).

References

The References field allows content creators to note the source material on which storyboard data is based on. This is typically a "production" field, and is not normally shown to the student.

User-Defined Field

Like Configuration, the User-Defined Field allows content creators to specify additional data. Use the *User-Defined Caption* textbox to specify the name of this field in the CCT.

Student Directions

The Student Directions field specifies directions associated with the storyboard (for example, "Click Next to continue").

Alternate Directions

Alternate directions are instructions associated with the storyboard that are only displayed under project-defined circumstances (for example, Alternate Directions could be displayed when the student has completed all required actions in the storyboard).

Narration Text

The Narration Text field is used to enter information for a human narrator or a text-to-speech system.

Narration File

The Narration File field allows the selection of an audio file containing narrated speech.

All Children Complete

If enabled, allows content creators to specify that any given storyboard should not be considered complete unless all of the children associated with that storyboard (indented beneath the storyboard within the lesson map) are marked as complete.

Remediation

The CCT allows content creators to use one of several different remediation approaches for question-type storyboards. The following remediation types can be enabled by checking the box next to the item.

Custom Messages

If enabled, this remediation type allows content creators to specify an individualized message for correct and incorrect responses for question types. Additionally, if the question type is a matching question, content creators can specify *extended custom messages*, individual messages for correct (and complete), correct but incomplete, partially correct, and totally incorrect responses to the set of matching question/answer pairs.

Storyboard Range

If enabled, this remediation type allows content creators to specify either a single or a range of storyboards to which the student should be remediated if their response is incorrect.

Matching Learning Objective ID

If enabled, this remediation type allows content creators to specify that the storyboards whose Objective ID value matches the question's Objective ID value should be used for remediation.

Per-Response Messages

If enabled, this remediation type allows content creators to specify messages for the correct answer and each individual distractor. When enabled, per-response messages are only available for multiple-choice storyboard types.

Supports Combined Remediation

If enabled, this option allows content creators to specify both a message-related remediation type and a storyboard range-related remediation type. The exact types available depend on which remediation types are enabled.

Type-specific properties

Several storyboard types, though not all, support type-specific options. Note that, as with all project properties, these settings affect the appearance of the CCT. Though the runtime is not typically affected not by these settings, your choices here may affect what data can be created with the CCT.

Type-specific property options

The following is a list of features supported in the type-specific tab, broken down by individual storyboard. Each storyboard's type-specific properties can be accessed from the *Storyboard Type* drop-down menu.

Title

Title-specific settings allow you to define whether or not the project supports additional text to be displayed on the Title screen.

Canvas

Canvas-specific settings include the size and color of the Canvas Editor's drawing area, background graphic sizing, default values for shape color, stroke width, and text color. Additional settings include action-specific color and stroke widths, as well as defining the allowed action types for a canvas element.

Flash Video

Flash Video-specific settings are available to set the size and color of the preview window, and whether or not the project supports stepped animations.

Rich Text

Rich Text-specific options are available to set the default size (width and height) of the Rich Text editor and font, the ability for the user to change fonts and/or font sizes, and whether or not the project supports background graphics.

Flash Object

Flash Object-specific options are available to set the size and color of the preview window and to allow entered data to be passed to the Flash .SWF.

Placeholder

Placeholder-specific options are available to define the supported placeholder types (this is a semicolon-delimited string) and whether or not the Placeholder Data field is visible.

Menu

Menu-specific options are available to define whether or not the project supports background graphics for menus and to define the maximum number of menu items supported.

Multiple-Choice Text

Multiple-Choice Text-specific options are available to define whether or not the project allows a supporting graphic to accompany a Multiple-Choice Text storyboard, and if so, to define the size and color of the graphic's preview widget.

Multiple-Choice Graphics

Multiple-Choice Graphics-specific options are available to set the size and color of the graphics.

Matching Text to Text

Matching Text to Text-specific options are available to define whether the project supports four matching question/answer pairs, or five.

Matching Text to Graphics

Matching Text to Graphics-specific options are available to define whether the project supports four matching question/answer pairs, or five, and to set the size and color of the preview widgets.

Importing and exporting project properties

Project Properties can be imported and exported to an XML file. Import an existing project properties XML file to reduce time and tedium.

How to import Project Properties:

- From Tools, select *Project Properties*. From File menu in Project Properties, select *Import Properties*. A standard Windows "Open" dialog appears.
- Choose the XML file you want to import from the Import Project Properties dialog window. Click *Open* to import the file. A confirmation dialog appears, indicating a successful import.

How to export Project Properties:

- From Tools, select *Project Properties*. From the Project Properties' File menu, select *Export Properties*. A standard Windows Save As dialog appears.
- Enter the filename into the save window that appears and choose *Save*. The project properties are saved to the named file, and a confirmation dialog appears, indicating a successful export.

How to reset default Property Values:

- From the Project Properties File menu, select *Reset Property Values*.

Chapter 5
previewer

Previewing

The Impression Content Creation Tool supports **previewing**—a review of created content within the actual runtime engine that will be used to deploy the content itself.

Previewing a storyboard.

Enabling previewing

You can enable previewing for a content database using the Project Properties. On the Main section of the Project Properties dialog, check the *supports preview* checkbox. Then, enter the fully qualified path and filename of the previewer in the *previewer application* text box.

How previewing works

When enabled, previewing adds two commands to the Lesson Editor (*Preview Lesson* and *Preview Storyboard*), and one to the Groups Window (*Preview Lesson*). These commands are available from the Tools menu of the given window.

When *Preview Lesson* is selected, the lesson's XML file is saved to the CCT user's temporary directory. This file and the media path associated with the lesson are passed as command-line arguments to the previewer application specified in the Project Properties.

If *Preview Storyboard* is selected, a lesson XML file is created from the specific storyboard, and that file is passed to the previewer application.

Tip: In both cases, once the CCT has launched the previewer, no further action is taken by the CCT; it is the responsibility of the previewer to display the lesson and to clean up the XML file generated by the CCT.

Creating a previewer

Without additional tools, Macromedia Flash-based applications do not have the ability to accept command-line parameters. As a result, you will typically need to create an application to receive the command-line parameters from the CCT and initialize the runtime based on those parameters. Impression includes a tool called the **Runtime Hosting Utility** that is designed to do just this. You can either use the Runtime Hosting Utility or create your own previewer.

Three command-line parameters will be passed to your custom previewer. The first is the fully qualified name of the XML file. The second is the media path. The third is the handle, called the *hWnd*, of the active CCT window, which is the parameter used by the Runtime Hosting Utility to center the previewer on the CCT.

Important notes:

Both the XML filename and the media path will be enclosed in double-quotes.

All path information is passed in Windows format. No URI information (e.g., "`file://`") is prepended to the paths. The previewing application is responsible for removing the XML file.

Runtime Hosting Utility

The Runtime Hosting Utility (RHU) is a lightweight application designed to provide a programmatic interface between the CCT and an RTE .SWF file. It was created specifically to

host content previewed from the CCT. The Runtime Hosting Utility executable, **RHU.EXE**, can be found in the CCT application directory. The VB6 source code for the RHU can be found on the *Runtime Engine Developer's Kit* CD-ROM.

Using the RHU

The RHU looks for an .INI file (**rhu.ini**) located in the same directory as **RHU.EXE**. This file contains all of the settings that the RHU uses.

To use the RHU for a specific content database, you should copy the **RHU.EXE** file to a centralized location accessible to all content creators.

Create the **RHU.INI** file (or use a copy of an existing file) to specify your project's RTE .SWF file and its size.

Finally, edit the content database's Project Properties so that the project supports a previewer, and specify the fully qualified path and filename to the RHU in the *previewer application* textbox on the Main section of the Project Properties dialog.

RHU.INI Settings

The following .**INI** settings control the RHU. All items are located in the **[shell]** section.

Setting	Description
width	Number. Specifies the stage width, in pixels, of the runtime engine being hosted.
height	Number. Specifies the stage height, in pixels, of the runtime engine being hosted.
file	String. Specifies the fully qualified path to the runtime engine .SWF.
debugMode	Boolean. When the runtime .SWF is loaded, the RHU will attempt to set the command object's *debugMode* variable to this value. If this entry is not found, the default value is *true*.
deleteXML	Boolean. If this value is true, the RHU will attempt to delete the XML file passed on the command line before the application is closed. If this entry is not found, the default value is true.

How the Runtime Hosting Utility works

The Runtime Hosting Utility (RHU) is designed to work with runtime engines that follow the same approach to initialization as the **MyShell** sample application; that is, the RTE must issue an *FSCommand* with a "*ready*" parameter when initialization is complete, and it must have an object named *command* available from the root level with *lessonXMLFile,*

mediaPath, and *action* variables. The action variable must support *loadLesson* as a setter value to start lesson loading.

The RHU itself begins by validating the XML filename and media paths passed on the command line, and the values it finds in the **`RHU.INI`** file. If any parameter is invalid, a message is displayed providing details of the error and processing stops. If all parameters are valid, the RHU resizes itself to the specified size and begins loading its RTE .SWF.

When the RTE is loaded and the *ready FSCommand* is received, the RHU passes the lesson data to the RTE by issuing a series of *.SetVariable* method calls on the internal Flash ActiveX control.

Before the RHU is closed, if the *deleteXML* value from RHU.INI is *true*, the RHU attempts to delete the XML filename passed as a command-line parameter. Note that this is a safe delete; the deleted file is placed in the Recycle Bin.

Appendix A
database validation

Impression database validation

When the Impression CCT opens a database connection, regardless of mode, a number of checks are performed to ensure that the database is valid, and in the case of SQL Server databases, that the user accessing the database has sufficient rights to access the database. If the CCT detects a problem, the opening process is cancelled, and a message is displayed on the Groups window.

Tip: Should an unexpected error occur at any time during the validation process, validation stops and the following message is displayed: *Impression encountered an unexpected error while attempting to validate [database name].*

Microsoft Access .MDB database checks

For a Microsoft Access .MDB file, the following checks are performed:

File existence

If the specified file does not exist, or no file is specified, one of the following messages is displayed:

- You must select a valid Impression database before using the program.

  —Or—

- [database name] was not found. Select a valid Impression database before using the program.

File is read-only

If the file exists, but is marked read-only, the following message is displayed:

- Impression was unable to obtain a shared read/write lock on [database name]. Check to ensure that the file is not marked read-only and that it is not stored in a directory with read-only permissions

Valid data connection

If a data connection to the file cannot be opened, the following message is displayed:

- Impression was unable to open a database connection to [database name]. This file may not be a valid Access-format database, or it may be corrupt. You may wish to validate the integrity of the file by opening it in Microsoft Access.

Writeable data structure

The CCT ensures that data fields can be edited by writing temporary data to the database, then removing that data. If an error occurs, the following message is displayed:

- Impression was unable to update one or more of the database tables. Check to ensure that no other processes have the database open exclusively and try again.

SQL connector database checks

When opening an SQL data source, the Impression connector database is validated first. Once the Connector passes validation, the specific content database is validated. Connector database validation does not include checking for writable data tables. If the account associated with the connector database does not have write permissions, the CCT itself will function normally, but users with administrator or superuser rights will not be able to use the ICMT to add or remove users or content databases from the Connector.

For the connector database, the following checks are performed:

Valid data connection string

If the data connection string (built from the Change SQL Connector dialog settings) is invalid, the following message is displayed:

- There is no data source or catalog defined for the connector. Please enter connection information and try again. Check with your administrator for details.

Valid data connection

Using the validated connection string, the CCT attempts to open a connection. If an error occurs, the following message is displayed:

- Impression was unable to open a database connection to the common connector. Check with your administrator to ensure that the machine and catalog are available and online, and that the account associated with the catalog has sufficient access rights.

Valid data structure

The CCT validates the data structure by running a series of SQL queries on the Impression-Specific tables in the database. If an error occurs, the following message is displayed:

- Impression was unable to locate all of the required database tables and fields. Contact your administrator for further information.

Connector record validation

The CCT checks to see if a record for the current user exists in the Connector database. If a record does not exist, one of the following messages is displayed:

If there are no user records in the database:

- The SQL connector database has no user records. If you are the administrator, use the Impression Connector Management Tool to initialize the database. If you are not the administrator, have the administrator perform this function.

If there are user records, but none for the current user:

- You do not have access rights to any database listed in the SQL connector. Contact your administrator for more information.

SQL Server content database checks

For the SQL Server-based content database, the following checks are performed:

Valid data connection

Using a connection string built from data in the Connector database, the CCT attempts to open a connection. If an error occurs, the following message is displayed:

- Impression was unable to open a database connection to [database name]. Check with your administrator to ensure that the machine and catalog are available and online, and that the account associated with the catalog has sufficient access rights.

Valid data structure

The CCT validates the data structure by running a series of SQL queries on the Impression-Specific tables in the database. If an error occurs, the following message is displayed:

- Impression was unable to locate all of the required database tables and fields. Contact your administrator for further information.

Writeable data structure

The CCT ensures that data fields can be edited by writing temporary data to the database, then removing that data. If an error occurs, the following message is displayed:

- Impression was unable to update one or more of the database tables. Check to ensure that no other processes have the database open exclusively and try again.

Appendix B
database security

Layered security

Layered security is the best way to protect your projects and databases. Security should be handled on a system-wide and user-wide level. Limiting user access to SQL Server machines and Connector user login information increases security. It only takes one ill-intentioned person with access to the SQL Server to corrupt the database. Limiting system-wide database use to LAN connections is another effective way to reduce the risk of breaches. Internet servers increase security risks and are not recommended for use with Impression.

SQL Server security procedures

As part of your SQL Server security procedures, implement maintenance and backup plans based on the criteria set by your organization. Implementing a maintenance and backup plan safeguards against data loss and optimizes database performance. Maintenance plans serve two purposes: data backup and providing statistical updates on data. Data replication allows for disaster recovery.

Microsoft Access security procedures

Access databases should be backed up frequently. Methods like backing up files with tapes and CDs can safeguard against data loss. Storing backups at a separate location provides added security against "natural disasters."

index