

STS Case Study Development Support

Dan A. Rosa de Jesús

JOHN F. KENNEDY SPACE CENTER

Major: Computer Engineering

USRP Spring Session

Date: 12 APR 2013

STS Case Study Development Support

Dan A. Rosa de Jesús¹

University of Puerto Rico, Mayagüez, Puerto Rico 00681

and

Grace K. Johnson²

John F. Kennedy Space Center, Kennedy Space Center, Florida 32899

The Shuttle Case Study Collection (SCSC) has been developed using lessons learned documented by NASA engineers, analysts, and contractors. The SCSC provides educators with a new tool to teach real-world engineering processes with the goal of providing unique educational materials that enhance critical thinking, decision-making and problem-solving skills. During this third phase of the project, responsibilities included: the revision of the Hyper Text Markup Language (HTML) source code to ensure all pages follow World Wide Web Consortium (W3C) standards, and the addition and edition of website content, including text, documents, and images. Basic HTML knowledge was required, as was basic knowledge of photo editing software, and training to learn how to use NASA's Content Management System for website design. The outcome of this project was its release to the public.

Nomenclature

<i>CDS</i>	=	Collection Display Size
<i>CMS</i>	=	Content Management System
<i>HTML</i>	=	Hyper Text Markup Language
<i>IPT</i>	=	Item Presentation Template
<i>JRE</i>	=	Java Runtime Environment
<i>KSC</i>	=	Kennedy Space Center
<i>NASA</i>	=	National Aeronautics and Space Administration
<i>Regex</i>	=	Regular Expression
<i>RSS</i>	=	Rich Site Summary
<i>SCSC</i>	=	Shuttle Case Study Collection
<i>UI</i>	=	User Interface
<i>XML</i>	=	Extensible Markup Language
<i>XSL</i>	=	Extensible Stylesheet Language

I. Introduction

The Shuttle Case Study Collection (SCSC) is a new educational resource of Kennedy Space Center (KSC). In an effort to capture and share knowledge gained from the Space Shuttle Program's 30-year history, the KSC Education Programs and University Research Division has developed the SCSC. Originally developed during summer and fall 2012, the mission of the SCSC is to provide educators with engineering case studies focused on Shuttle maintenance, processing, and operations. This will give students real world experiences to encourage the development of decision-making, critical thinking, and problem-solving skills.

Official lessons learned from NASA programs and projects are used as baselines for the case study reports. Due to the common brevity of lessons learned documents, further research and development is required to expand them into

¹ Intern, Education Programs & University Research Division, Kennedy Space Center, University of Puerto Rico, Mayagüez Campus.

² Technical Project Manager, Education Programs & University Research Division, EX-E, Kennedy Space Center.

case study reports. These reports provide additional details and further understanding about the related Shuttle mission, its crew, unique system anomalies, testing, or design modifications as applicable. Educators may incorporate these reports, with accompanying epilogue and teaching notes, as a new tool to allow their students to participate in classroom discussions, group activities, and research projects. Through the case study reports, students are challenged to think like official NASA team members responsible for making critical decisions related to the Space Shuttle Program.

For spring 2013, the development of the SCSC continued with the development of its website. This research project constitutes the creation of a website to host the SCSC that requires a database to store the case study collection, an option to allow online users such as high school educators, university professors, and NASA civil servants to submit their own case studies to the collection for review and the capability of hosting image, videos, and many other multimedia content. The website allows the distribution of case study materials to the public as well as communication between case study users and the collection's management. Since this was the final stage of the SCSC development, it was essential for the website to be user-friendly for maximum usability, elegant and clear to understand by its targeted audience, and to follow the World Wide Web Consortium (W3C) standards for its functionality in environments that provide their required features. The website is also required to comply with NASA regulations while integrated with other education sites included within the NASA Portal. To meet these requirements, NASA's Content Management System (CMS) is used.

II. Content Management System & Its Assets

Creating and editing content on a NASA website means publishing to the NASA Portal. It requires NASA Portal publishing training which includes how to use NASA's CMS and its capabilities to manage digital files, apply workflows to publish pages, and ensure the 508 compliance to make content accessible for broad audiences. NASA's CMS provides audiences with a single point of entry to NASA material resources with the most up-to-date content allowing easy navigation through the NASA Portal, a consistent appearance, quality content to create a unique user experience, and security to reduce high cost of security investigations. To ensure the Portal's success, NASA's content managers should understand the needs of the internal and external NASA constituents, to ensure that the content promotes NASA's vision, commit to provide timely and compelling content to their websites, and ensure a consistent user experience¹.

NASA's CMS has two main types of assets, documents and libraries. Document assets are considered structured assets that consist of properties, content, promotional content, collections, and metadata, while library assets are unstructured pieces of content. Document assets are created to develop the bulk of content for a page or collection. Library assets are commonly used for multimedia objects such as images, videos, links, and downloadable content, including PDFs.

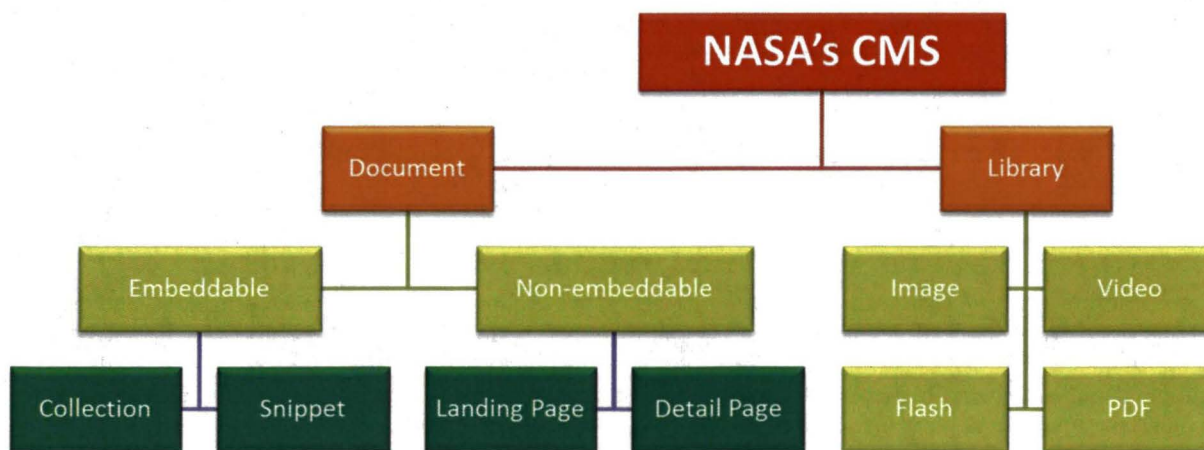


Figure 1. NASA's CMS General Structure. Note that documents can be divided into two sub-types, embeddable (can be inserted into other documents as additional resources) and non-embeddable¹.

Assets that are going to be created for the website require a hierarchical category within NASA's CMS. Hierarchical categories work in the same way folder directories work. Since the SCSC website is under the

Education Office and it is being developed at KSC, assets are stored under the following category: Root > NASARoot(Production) > Content > offices > education > centers > kennedy > technology. Categories within NASA's CMS conform to the NASA Portal Information Architecture, which is required for NASA's published websites. Correct storage of assets in NASA's CMS hierarchical category helps to ensure that published pages are displayed with the appropriate navigation options. Once the proper category is selected, assets may be created accordingly.

Before the website can be published, some steps are required:

1. Creating library assets
2. Creating document assets
3. Creating collection of collections or a collection of documents
4. Creating a landing page
5. Applying workflow to appropriate assets

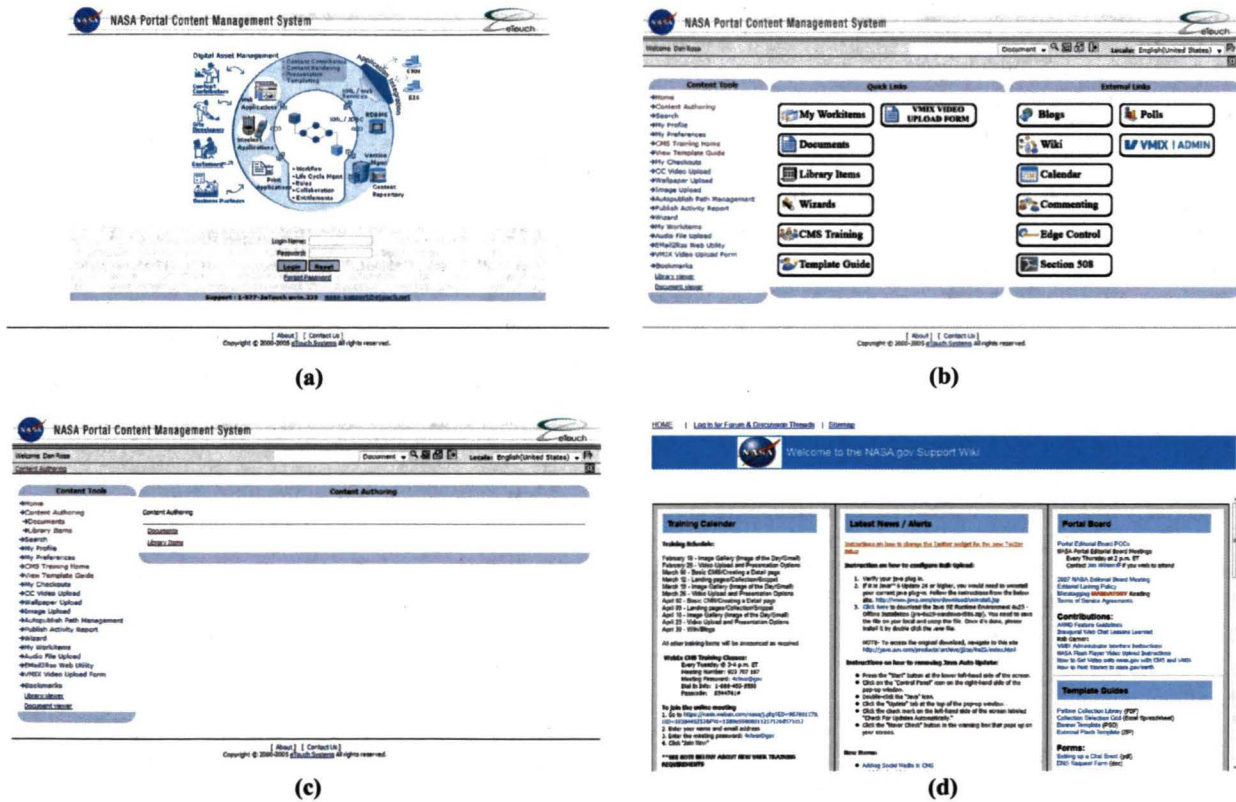


Figure 2. NASA's CMS Navigation Options. The login (a), home (b), content authoring (c), and CMS training pages.

III. Creating Document Assets

Documents are structured assets that consist of properties, content, promotional content, collections, and metadata. The most important feature of document assets is that they can be transformed into publishable, light Hyper Text Markup Language (HTML) pages. In most cases their implementation will require a presentation template that will define the document's user interface (UI) appearance and the order in which its content will appear on the screen, including text, images and other media objects.

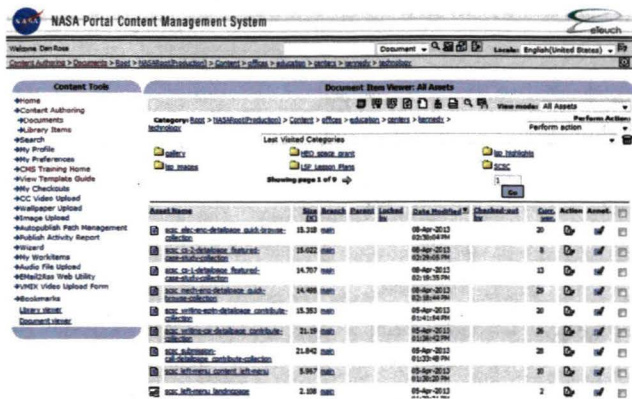


Figure 3. Document Assets Content Authoring Section. NASA's CMS provides a feature to sort assets by name, size, date modified, checked-out by, and version.

Depending on the selected combination of authoring template, presentation template, and document family properties to create the new document asset, NASA's CMS will show different arrangements of properties to be set for the document to work properly when it is integrated with other assets. Those properties will be described later in this document.

IV. Creating Library Assets

Library assets are reusable and unstructured pieces of content commonly used for multimedia objects such as images, videos, links, and downloadable content like PDF files. Sometimes, multimedia objects may need to complete some pre-processing steps, specifically for images. Images usually require pre-sizing or editing using a graphics editing program, such as Adobe Photoshop, before creating an image asset in NASA's CMS. For example, to meet the promotional thumbnail image requirements, images must be 226 pixels wide and 170 pixels high¹. Keep in mind that other types of multimedia objects may require text, audio, and video editing. These pre-processed media objects may then be uploaded to NASA's CMS Library Items section under the Content Authoring tool.

Once the multimedia objects are pre-processed, the first step for creating a library asset is to define its properties. The name of a library asset must have no spaces or special characters with the exception of underscores and dashes as indicated for document assets. Asset types such as file, flash, image, or link should be selected. As mentioned for document assets, keywords and a description of the library asset should also be defined within the properties during creation. NASA's CMS provides a bulk upload option for multiple items³. Unfortunately, the setup and use of the bulk upload option requires an older version of Java Runtime Environment (JRE), setting its options to not to auto update, and the request of IT elevated privileges. Once property definitions are set and the "Create" button is pressed, NASA's CMS will allow editors to proceed with content creation, the second step for creating library assets.

During content creation, editors may browse their computer to locate the media objects to be uploaded. Additional properties may also be required under the "Content" tab. In particular, to conform to the 508 Compliant Rules and Regulations which govern the web for disabled individuals, all images must include an alternate text property. After

The setup of a series of properties is required when creating document assets. These include the name, authoring template, presentation template, document type, document family, Extensible Stylesheet Language (XSL), document, document template, HTML, HTML template, asset reference, search agent, and doclet. Keywords and a description of the document asset should also be defined within the properties during creation. Keywords are the words that describe the internal significance of the document asset and are commonly used for metadata search, a more specific search provided by NASA's CMS. The description is a statement that provides a concise, human-readable summary of each document asset. After these properties are set and the "Create" button is pressed, NASA's CMS will allow editors to proceed with content creation of the document asset.

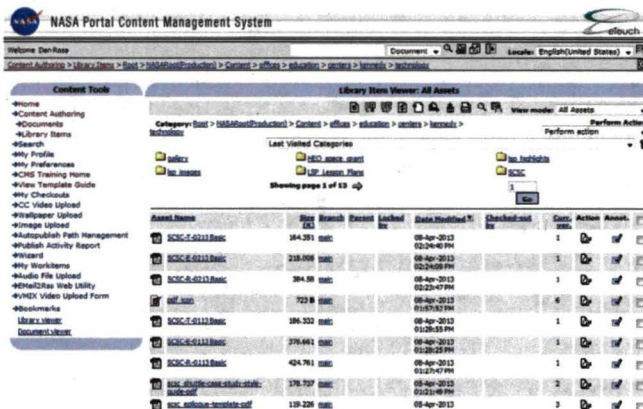


Figure 4. Library Assets Content Authoring Section. Each asset is assigned with an icon that describes its type.

images of different sizes, and the leader sentence. The promotional title should indicate the topic of a detail page, thumbnails should show a picture related to the information of a detail page, and the leader sentence should be a summary of the information contained in a detail page. The promotional content will only be shown when a detail page is embedded in a collection. Depending on the collection in which a detail page will be embedded, editors may decide which of the promotional content properties to use. This matter will be discussed shortly.

C. Collections

NASA's CMS allows document assets such as detail pages and snippets along with library items to be combined to create collections. Just like document and library item assets, the creation of a collection requires the definition of a set of properties, including name, authoring and presentation template, document type, and document family. With these properties set, the collection content is ready to be added.

There are very noticeable differences between the "Content" of collections and other document assets. Because collections consist of a combination of documents assets, there is no body property for them. This means that detail pages and snippets may be embedded within a collection through the asset reference property by browsing NASA's CMS categories.

Another difference between collections and other document assets is the template property. In addition to the authoring and presentation templates set under the "Properties" tab, collections also require setting an Item Presentation Template (IPT) property under the "Content tab". One of the most important aspects of collection creation is ensuring that the IPT matches the Collection Display Size (CDS). Formatting options of these properties must agree for text and images to display properly. For example, if the IPT is selected to display a collection for blind users, the CDS must be selected to display the content adjusted for blind users. In cases where the IPT includes a video or other multimedia objects, so must the CDS. Additionally, unlike detail pages, where content could be edited and previewed accordingly, collections cannot be previewed. Editors must save the collection and embed it in the document asset desired. In other words, editors must "Check-In" the collection and develop a landing page or detail pages in order for the collection to be properly previewed. This makes sense since collections and library assets require no publication, when a website is being released to the public. If the collection along with its content is not adequately shown during the preview of the container document asset, the collection must be "Checked-Out" so modifications can be made to edit the collection's appearance. Once changes are complete, the collection must be checked in again before previewing the container document asset. No collection can be previewed from the container document asset unless it is "Checked-In" first. "Check-In" and "Check-Out" actions will be explained later in this document.

Depending on how content is going to be displayed, editors may decide which promotional content properties to set on their document and library assets to then embed them in a collection. For example, the "Quick Browse" collection requires a promotional thumbnail 226 pixels wide and 170 high. This means that there is no necessity of setting any of the other promotional image properties unless the IPT of the collection is changed.



Figure 6. Collections & Its Role. Collections function as a bridge between the landing page and the document and library assets. The idea behind this is to combine the assets through a collection. Then, the collection is integrated to the landing page.

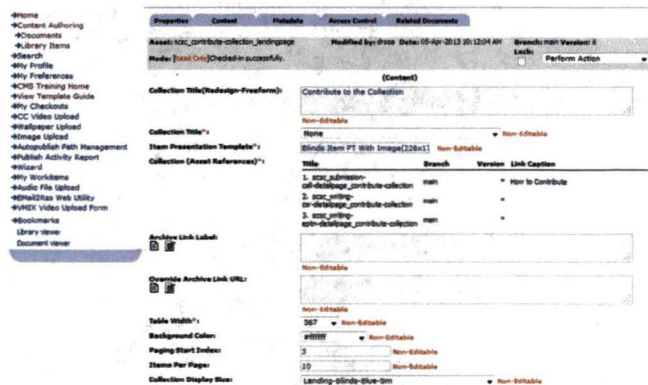


Figure 7. Collections Content Section. Multiple document assets, like snippets and detail pages, can be added to a collection through the asset references property provided by in this section.

D. Landing Page

With detail pages and snippet collections created, the next step in developing the SCSC website is the creation of a landing page. At this point of the SCSC website development, collections may be embedded in the landing page for displaying its content. Landing pages can have up to twelve separate collections consisting of combinations of linked document and library assets¹. The SCSC website requires a single landing page.

To create a landing page, another set of properties is required as with document and library assets, including the authoring and presentation templates, document type and document family, keywords, and description. Different from detail pages and snippets, there is no body property provided for landing pages. However, it provides a set of properties to embed collections depending on where their placement is desired including top, bottom, right, left, or center¹. Additional promotional properties are also available.

For the SCSC, several collections are included within the landing page. For example, the “Introducing the Shuttle Case Study Collection” collection was created to introduce and familiarize the audience with the website’s content. It required the implementation of a snippet that was then referenced to a collection embedded in the landing page. A detail page was created for each engineering discipline in which links to each related case study is embedded within its body. These separated detail pages were then selected under the asset references property for collections to include them in the “Quick Links” collection. In general, collections contain different combinations of document and library assets for display on the landing page.

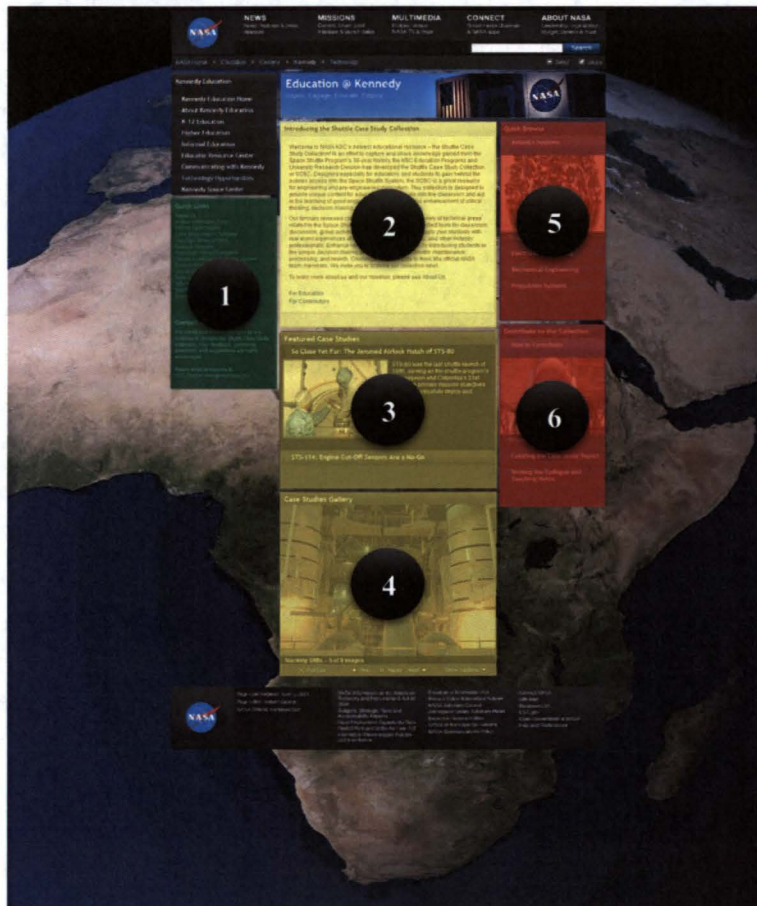


Figure 8. SCSC Landing Page. The SCSC landing page implements a template of three columns with room for up to twelve collections. The green shaded area is the left navigation menu collection that contains the quick links and contact snippets. The yellow shaded area includes the introduction (2), featured case studies (3), and case study gallery collections (5). Finally, the red shaded area features the quick browse (5) and contribute to the collection collections (6).

VI. Creating an Image Gallery



Figure 9. SCSC Image Gallery. NASA's CMS auto generates a control panel for the image gallery collection. It gives the audience an option to see each picture in full screen mode, go back and forward through the images, play and pause slide shows, and read captions.

One of the most challenging aspects of developing a website through NASA's CMS is the creation of an image gallery. It requires the implementation of five different types of document assets and at least one image asset. These are the image gallery, image gallery collection, image gallery Rich Site Summary (RSS), image gallery module, and image gallery module collection. The parent of all these assets is the image gallery module collection and its hierarchical structure is the following: image gallery module collection > image gallery module > image gallery RSS > image gallery collection > image gallery > images. Most of these assets are wrappers that complement each other with a specific function for the image gallery to work properly. However, the image gallery RSS plays an important role in the image gallery functionalities. It contains metadata that tells the

image gallery where to find the image assets in NASA's CMS database.

Another interesting aspect of developing image galleries is that they cannot be previewed even if its assets are checked in. To preview the image gallery, editors have to publish to stage the document asset that contains the image gallery, causing difficulty and slowing down the website development process.

As a final point, the promotional content of each image asset embedded in the image gallery should be completed in its entirety, since NASA's CMS generates a menu to access a larger version of those images along with their complete description, as well as links for downloading them.

VII. Check-In, Check-Out, Update, Delete, & Move Actions

The concepts of checking in, checking out, and updating document and library assets have been previously mentioned in this document. NASA's CMS keeps track of the version of document and library assets. When an asset is created, NASA's CMS sets its version to null. After a "Check-In" action is made on an asset, NASA's CMS will automatically increase its version, keeping the previous one saved in the database. To return to a previous version of an asset, editors will have to use the search feature provided by NASA's CMS. If the searching yields results, the desired version of the asset may be selected to then check out to make changes and check in to override its last version. Even so, "Check-In" also implies a more intuitive action, that the asset is ready for publication.

In some cases the versioning tool of NASA's CMS is misleading. For example, there are times where editors want to make a minimum edition to an asset. It could be a simple modification to one of its properties or the correction of a typo in its body property. In that situation, updating the asset is preferred rather than checking it in since the last one will not represent a significant difference from one version to another. The landing page and detail pages follow this rule. However, collections and library assets are the exception. This implies that those types of assets must be checked in regardless of the modification made to them for previewing its content, since they require no publication when a website is being released to the public.

As stated earlier in this document, the "Check-Out" action may be used by editors to start modifying document and library assets. Nevertheless, an asset is no ready for publication, when is checked out. Actually, a checked out asset cannot be released to the public. The "Check-Out" action is only enabled for use if the asset is already checked in.

Sometimes editors may want to delete assets. Although this is a common action offered by many operating systems, other CMSs and mobile applications, it could be tricky to perform when using NASA's CMS. Unexpectedly, an error could pop-up with a message stating that the asset or assets selected cannot be deleted. The only workaround to this issue is moving the desired asset or assets to the recycle bin category provided by NASA's CMS under Content Authoring > Documents > Root > NASARoot(Production) > Content > RecycleBin. Selecting the "Move" action will pop-up a window to select where to move the asset or assets selected. If the asset or assets to

be moved already exist in the targeted category, NASA's CMS will ask to rename them by itself or cancel the action.

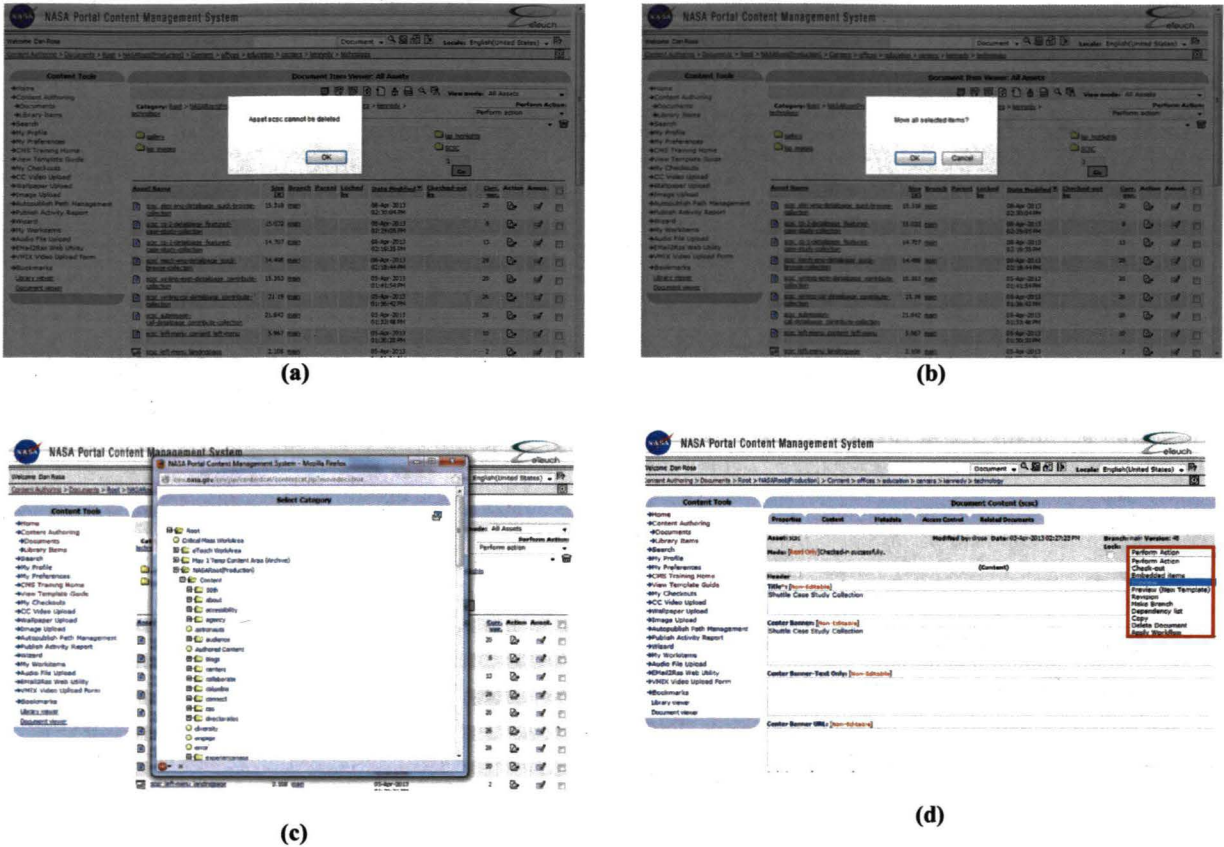


Figure 10. NASA's CMS Actions. A message triggered when an asset cannot be deleted (a). Dialog box requesting approval for the movement of an asset to another category (b). A window with a list of categories where the selected assets will be possibly moved (c). Actions provided by NASA's CMS (d).

VIII. Document & Library Assets Name Nomenclature

After the first two phases of the SCSC website development, the amount of documents and library assets in NASA's CMS database became robust making it difficult to differentiate SCSC assets from other website's assets and understand the relationship between the SCSC website assets. To solve that issue, a name nomenclature was designed following two main rules. An asset must have a name with no spaces or special characters with the exception of underscores and dashes and cannot exceed the limit of eighty characters.

Although the nomenclature worked well with these rules, it lacked of a way of filtering the SCSC assets from other assets, when using NASA's CMS search tool. It also required a way to link the name of an asset with its parent asset, the container document asset. Most of the assets belonging to the SCSC website are named using the nomenclature composed by the following pattern:

`scsc_[asset-name]_[parent-asset-name]`

The "scsc" string is required at the start of each SCSC asset name along with the underscores separating each component in the pattern. The second component of this pattern is the asset name with letters and numbers, separated by dashes. The last component is the parent asset name, the container asset, with letters and numbers also separated by dashes. The "scsc" string is used for searching assets through NASA's CMS search tool and the other two components are used for linking assets within SCSC website. The generalization of this pattern can be written as the following Regular Expression (RegEx):

scsc_[a-zA-Z][a-zA-Z0-9-]{0,36}_[a-zA-Z][a-zA-Z0-9-]{0,36}

A regular expression is a specific pattern that provides concise and flexible means to "match" strings of text, such as particular characters, words, or patterns of characters⁴. In this case, "a-zA-Z" matches a lowercased or uppercased letter from 'a' to 'z', "a-zA-Z0-9-" also matches lowercased and uppercased letters from 'a' to 'z', but includes numbers from '0' to '9' with dashes, and "{0, 36}" matches the repetition of the previous token ("a-zA-Z0-9-") from zero to thirty-six times. Asset name examples matching this RegEx are "scsc_intro-detailpage_landingpage", "scsc_left-menu_left-nav-collection", and "scsc_image4_226x170_study-case-1".

A good example of the SCSC name nomenclature in action is the implementation of a snippet to show some content in a landing page. This requires the creation of three document assets including the snippet with the text, HTML tags and other embedded library assets, a collection to wrap the snippet, and, of course, the landing page to embed the collection. The names of the assets would be the following:

Document Asset Type	Asset Name
Snippet	scsc_case-study1-snippet case-study1-collection
Collection	scsc_case-study1-collection landingpage
Landing Page	scsc landingpage

From the snippet point of view, "case-study1-snippet" represents its name and "case-study1-collection" is the name of the document asset where it is embedded. On the other hand, "case-study1-collection" is the name of the collection that contains the snippet and "landingpage" is the name of the document asset where it is embedded. Finally, the "landingpage" is the name of the root parent of the other two assets. In general, it can be said that the snippet is inside a collection that is inside the landing page. This way enhances how the document and library assets are handled by editors in terms of their hierarchical relationship through an intuitive and organized method.

IX. Conclusion

In an effort to provide high school and university educators with Shuttle-related and additional engineering materials, the Shuttle Case Study Collection has been developed. Since summer 2012, the first and second phases of this project have been focused in completing the general design and content of its website. During this third phase of the project, experiences from those two previous phases were used to accelerate the edition and addition of content to the SCSC website. With that in mind, a new way of naming the document and library assets was introduced with the intention of helping other editors to understand, through an intuitive way, the relationship between assets. Other experiences gained during this internship opportunity are shared in this document including specific information about the creation of document and library assets, assets needed for creating an image gallery and their functionalities, importance of checking in and checking out assets, and guidance for solving some of NASA's CMS uncommon issues found. After almost a year of hard work, the SCSC website is finally released to the public. It may be accessed through <http://www.nasa.gov/offices/education/centers/kennedy/technology/scsc.html>.

Acknowledgments

Special thanks to the Undergraduate Student Research Program and all the parties that made this internship opportunity possible. Special thanks to Rob Cannon (NASA KSC) for his tireless support with NASA's CMS understanding and troubleshooting. Special thanks to Grace K. Johnson (NASA KSC) for her unique mentoring. Finally, thanks to all the people that somehow helped me to gain new skills for stepping forward professionally.

References

¹eTouch Systems Corp., "NASA Portal Content Management System (CMS) Training", Fremont, California, February, 2008. [http://support.nasa.gov/1000859main_CMS%20Training%20Document%20Version5_02-2008.ppt. Accessed 4/8/13.].

²eTouch Systems Corp., "How to Create a Detail Page in the Content Management System (CMS)", Fremont, California, January, 2010. [http://support.nasa.gov/1011650main_documentation-detailpage.pdf. Accessed 4/8/13.].

³eTouch Systems Corp., “How to Upload Images to the Library Asset Area in the Content Management System (CMS)”, Fremont, California, June, 2009. [http://support.nasa.gov/1003665main_documentation-imageupload.pdf. Accessed 4/8/13.].

⁴Wikimedia Foundation, Inc., “Regular expression”, San Francisco, California, April, 2013. [http://en.wikipedia.org/wiki/Regular_expression. Accessed 4/8/13.].