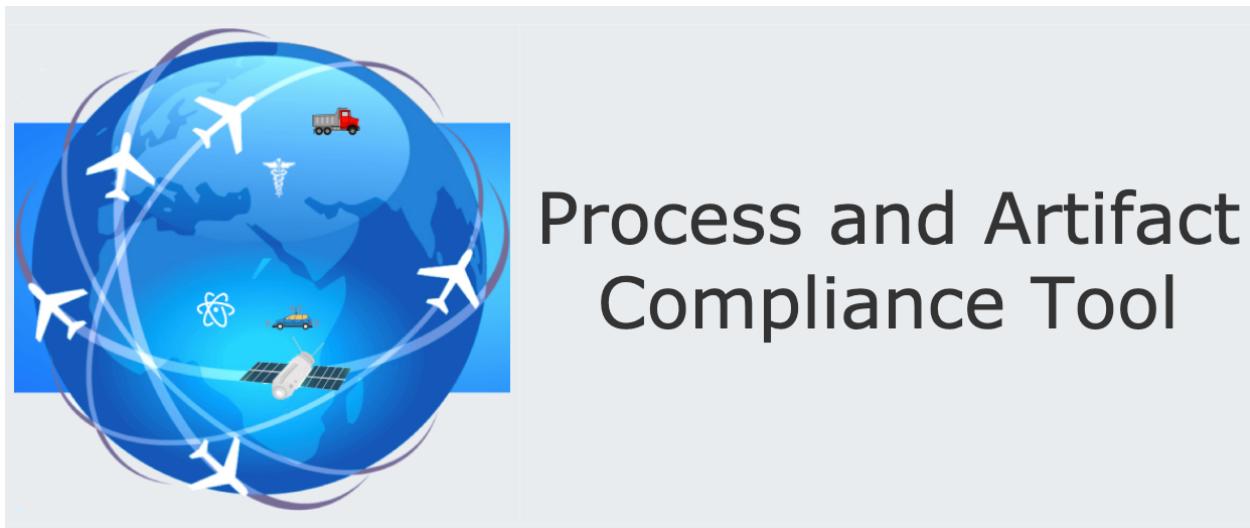




User Guide

Version 1.4 19 August 2021



Airworthiness Certification Services, LLC

PMB#224, 5500 Olympic Drive Suite H-105, Gig Harbor, WA 98335

Phone/fax: 425-427-1956

support@Airworthinesscert.com

<https://www.airworthinesscert.com/>

© 2020 Airworthiness Certification Services (ACS). All rights reserved.

Trademarks

PACT is a registered trademark of ACS. All other trademarks or registered trademarks are the property of their respective owners.

Disclaimer

The information provided in this document is provided "as is" without warranty of any kind. ACS disclaims all warranties, either express or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall ACS be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if ACS or its suppliers have been advised of the possibility of such damages.

Technical Requirements

PACT will work with the following Operating Systems: Windows 7, 8, 10 and above, Mac OSX Sierra and above, Linux Ubuntu 18.x and above, Chrome OS 79 and above.

PACT will work with the following Browsers: Internet Explorer 11 and above, Microsoft Edge 16 and above, Safari 11 and above, Chrome Version 62 and above, Firefox 68 and above, Opera 45 and above.

Product information & documentation

Documentation, release notes, software updates, and information about ACS products, licensing, and service, are at the ACS website:

<https://www.airworthinesscert.com/pact-tools/downloads/>

ACS may occasionally update online documentation between releases of the related software. Consequently, if this document was not downloaded recently, it may not contain the most up-to-date information. You may download and refresh this document if it has been updated, as indicated by a change in this date: August 10, 2020

Technical support

Email support@Airworthinesscert.com. Note that to open a service request, you must have a valid support agreement.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send comments and/or your opinion of this document to:

info@Airworthinesscert.com

If you have issues, comments, or questions about specific information or procedures, please include the title and, if available, the part number, the revision, the page numbers, and any other details that will help us locate the subject that you are addressing.

Preface

Style Conventions

The following style conventions are used in this document:

Normal: This type of text is used in general descriptions.

Initial Capitals: Initial capital letters indicate an important term, an element that has significance in PACT, or a heading.

“Quotes”: Quotes are used when introducing new terms.

Bold: This is the type of text used to identify a PACT Page as well as Fields or areas within a page. A Field is an area that a user fills in on a PACT Page. A Page is a user interface window (i.e., screen or view) within which the User enters and/or receives data.

Italic: Italics are used for publication titles referenced in text, emphasis, and the introduction of new terms.

Courier: This type of font is used for URLs, complete paths, filenames, file formats, prompts, code and syntax



These symbols designate either 1) notes to the user, usually indicating something that is not intuitive or that must be understood about an operation or its results, 2) user tips, or 3) highlighting a feature that is not intuitive or visible.



These symbols provide warnings, usually indicating that an operation may have some serious consequences and should be performed only under special or controlled circumstances.

Table of Contents

1	<i>Introduction to PACT</i>	11
1.1	What Does PACT Do?	11
1.2	Understanding the PACT Graphical User Interface	12
2	<i>PACT Usage Overview</i>	17
2.1	Setting Up Your Account	17
2.2	Logging In	17
2.3	Changing Your Password	19
2.4	Listing PACT Projects	20
2.5	Setting up Your First Project	20
2.6	Setting Up Hardware/Software Items	24
2.7	Using PACT Through the Development Life Cycle	30
3	<i>Defining Team Members and Roles</i>	32
3.1	User Roles	32
3.2	User Role Scope	34
3.3	Administrator Role	35
4	<i>Capturing and Managing Requirements</i>	39
4.1	Essential Elements of Requirements Management	39
4.2	Setting Up System Requirements	39
4.3	Managing Hardware/Software Item Requirements	51
4.4	Managing Module Descriptions	63
4.5	Requirements Traceability	75
5	<i>Managing Documentation</i>	76
5.1	Documentation Throughout a Life Cycle	76
5.2	Understanding PACT Versioning	78
5.3	Document Templates	78
5.4	Listing the Documents	87

5.5	Viewing a Document	88
5.6	Downloading a Document	89
5.7	Creating a New Document	89
5.8	Uploading a Document	91
5.9	Deleting Documents	91
5.10	Viewing Document History	92
5.11	Commenting on a Document	92
5.12	Reviewing a Document	94
6	<i>Managing Source Code</i>	95
6.1	Creating a New Source Code Record	96
6.2	Editing a Source Code Record	97
6.3	Deleting Source Code	98
6.4	Linking Source Code to Requirements	98
6.5	Viewing Source Code	100
6.6	Viewing Source Code History	101
6.7	Exporting Source Code	103
6.8	Importing Source Code	103
6.9	Zip Source Code Files	104
6.10	Scanning GitHub or GitLab Files	105
6.11	Renumbering Source Code	108
6.12	Baselining Source Code Records	109
6.13	Viewing Source Code Baselines	110
6.14	Utilizing Source Code Coverage	112
7	<i>Managing Test Cases & Test Procedures</i>	122
7.1	Understanding Test Cases and Test Procedures	122
7.2	PACT Support of Test Cases & Procedures	122
7.3	Managing Test Cases in PACT	123
7.4	Managing Test Procedures in PACT	130

8	<i>Managing Model Files</i>	137
8.1	Creating a New Model File Record	138
8.2	Editing a Model File Record.....	139
8.3	Deleting a Model File	139
8.4	Linking a Model File to Requirements	140
8.5	Exporting a Model File	140
8.6	Importing a Model File.....	141
9	<i>Establishing Requirements Traceability</i>	143
9.1	Requirements Traceability Matrix	143
9.2	Viewing the RTM.....	145
9.3	Exporting the RTM	146
10	<i>Holding Reviews</i>	147
10.1	Understanding Reviews	147
10.2	PACT Review Flow.....	149
10.3	Project Review Status	151
10.4	The Review Process for Hardware/Software Items.....	154
10.5	Setting Up Checklists.....	155
10.6	Setting Up a New Review	156
10.7	Joining a New Review	159
10.8	Participating in a Review.....	160
10.9	Tracking and Resolving Issues from a Review	165
10.10	Passing a Review	170
10.11	Exporting Checklists	171
11	<i>Managing Problem Reports</i>	172
11.1	Understanding the Problem Reporting Flow.....	172
11.2	Listing Problem Reports	174
11.3	Creating a New Problem Report.....	177
11.4	Editing a Problem Report	182

11.5 Viewing a Problem Report	184
11.6 Emailing Problem Reports.....	185
11.7 Exporting Problem Reports	185
11.8 Importing Problem Reports.....	186
11.9 Viewing Problem Report History	187
12 Producing Certification Paperwork.....	189
13 Archiving Your Project Data	191
13.1 Understanding PACT Archiving.....	191
13.2 Exporting Documentation & Artifacts.....	191
13.3 Archiving Project Data & Tool	192
13.4 Onsite Use and Archiving	195
Appendix A: Acronyms.....	196
Appendix B: Glossary of Terms.....	200
Appendix C: Roles & Access.....	206

Table of Figures

<i>Figure 1-1. PACT Home Page</i>	12
<i>Figure 1-2. PACT Home Page Icon</i>	12
<i>Figure 1-3. PACT Projects List Page</i>	13
<i>Figure 1-4. Menu Bar.....</i>	13
<i>Figure 1-5. Action Buttons</i>	15
<i>Figure 2-1. Login Page</i>	18
<i>Figure 2-2. Change Password Menu Item</i>	19
<i>Figure 2-3. Change Your Password Page</i>	19
<i>Figure 2-4. New Project Page.....</i>	21
<i>Figure 2-5. Project Page.....</i>	23
<i>Figure 2-6. Project/Item Hierarchy in PACT</i>	25
<i>Figure 2-7. Hardware/Software Items List Page</i>	26
<i>Figure 2-8. New Item Page</i>	27
<i>Figure 2-9. Specific Hardware/Software Item Page.....</i>	29
<i>Figure 3-1. Administrator Menu</i>	35
<i>Figure 3-2. New (or Edit) User Page.....</i>	36
<i>Figure 4-1. System Requirements List Page</i>	40
<i>Figure 4-2. Specific System Requirement Page</i>	41

<i>Figure 4-3. New System Requirement Page</i>	42
<i>Figure 4-4. System Requirements Allocation (i.e. Requirements Tracing) Page</i>	44
<i>Figure 4-5. System Requirements Not Outlinked</i>	45
<i>Figure 4-6. Import System Requirements Page</i>	46
<i>Figure 4-7. Export System Requirements Page</i>	46
<i>Figure 4-8. New System Requirements Baseline Page</i>	48
<i>Figure 4-9. System Requirements Baseline List Page</i>	49
<i>Figure 4-10. Viewing a Specific Systems Requirements Baseline</i>	50
<i>Figure 4-11. High-Level Requirements List Page</i>	52
<i>Figure 4-12. New High-Level Requirement Page</i>	53
<i>Figure 4-13. Link System Requirements Panel</i>	55
<i>Figure 4-14. Link High-Level Requirements Panel</i>	56
<i>Figure 4-15. Export High-Level Requirements Page</i>	57
<i>Figure 4-16. Import High-Level Requirements Page</i>	58
<i>Figure 4-17. New High-Level Requirements Baseline Page</i>	60
<i>Figure 4-18. High-Level Requirements Baselines List Page</i>	61
<i>Figure 4-19. Viewing a Specific High-Levles Requirements Baseline</i>	62
<i>Figure 4-11. Module Descriptions List Page</i>	64
<i>Figure 4-12. New Module Description Requirement Page</i>	65
<i>Figure 4-13. Link High-Level Requirements Panel</i>	67
<i>Figure 4-13. Link Low-Level Requirements Panel</i>	68
<i>Figure 4-15. Export High-Level Requirements Page</i>	69
<i>Figure 4-16. Import Module Definitions Requirements Page</i>	70
<i>Figure 4-17. New High-Level Requirements Baseline Page</i>	72
<i>Figure 4-18. High-Level Requirements Baselines List Page</i>	73
<i>Figure 4-19. Viewing a Specific High-Levles Requirements Baseline</i>	74
<i>Figure 5-1. Documentation Flow</i>	76
<i>Figure 5-2. Template Scope and Flow</i>	80
<i>Figure 5-3. Administrator Access to Templates</i>	81
<i>Figure 5-4. Templates List Page</i>	81
<i>Figure 5-5. Accessing ACS Document Templates</i>	85
<i>Figure 5-6. ACS Template Documents List Page</i>	85
<i>Figure 5-7. Editing Template Document Page</i>	86
<i>Figure 5-8. Organization Template Documents List Page</i>	87
<i>Figure 5-9. Documents List Page</i>	88
<i>Figure 5-10. New Document Page</i>	90
<i>Figure 5-11. Document History Page</i>	92
<i>Figure 5-12. Document Comments</i>	93
<i>Figure 5-13. Document Comments List Page</i>	93
<i>Figure 5-14. New Document Comment Page</i>	94

<i>Figure 6-1. Source Code List Page</i>	95
<i>Figure 6-2. New Source Code Page</i>	96
<i>Figure 6-3. Link Source to Low-Level Requirements</i>	98
<i>Figure 6-4. View Source Code Page</i>	100
<i>Figure 7-1. Test Cases List Page</i>	123
<i>Figure 7-2. Viewing a Specific Test Case</i>	124
<i>Figure 7-3. New Test Case Page</i>	125
<i>Figure 7-4. Link Test Cases to High-Level Requirements</i>	127
<i>Figure 7-5. Export Test Cases Page</i>	128
<i>Figure 7-6. Import Test Cases Page</i>	129
<i>Figure 7-7. Test Procedure List Page</i>	131
<i>Figure 7-8. Showing Test Procedure Information</i>	131
<i>Figure 7-9. New Test Procedure Page</i>	132
<i>Figure 7-10. Link Test Procedures to Test Cases</i>	134
<i>Figure 7-11. Export Test Procedures Page</i>	135
<i>Figure 7-12. Import Test Procedures Page</i>	136
<i>Figure 8-1. Model File List Page</i>	137
<i>Figure 8-2. New Model File Page</i>	138
<i>Figure 8-3. Export Model File Page</i>	141
<i>Figure 8-4. Import Model File Page</i>	141
<i>Figure 9-1. Requirements Tracing Page</i>	144
<i>Figure 9-2. Reversed Requirements Tracing Page</i>	144
<i>Figure 9-3. Requirements Tracing Example</i>	145
<i>Figure 9-4. Export Requirements Tracing Matrix Page</i>	146
<i>Figure 10-1. Typical DO-254 Program Reviews</i>	148
<i>Figure 10-2. Typical DO-178C Program Reviews</i>	149
<i>Figure 10-3. Review Flow in PACT</i>	150
<i>Figure 10-4. Project-Level Review Status Page</i>	152
<i>Figure 10-5. Specific Review Status Page</i>	153
<i>Figure 10-6. Checklist Associated with a Review</i>	154
<i>Figure 10-7. Item-Level Reviews List Page</i>	155
<i>Figure 10-8. New Review Page</i>	157
<i>Figure 10-9. Email to Evaluators</i>	158
<i>Figure 10-10. Information Portion of Editing Review Page</i>	159
<i>Figure 10-11. Checklist Portion of Editing Review Page</i>	160
<i>Figure 10-12. Edit User Info Menu Item</i>	161
<i>Figure 10-13. Editing User Page</i>	162
<i>Figure 10-14. Sign-In Sheet</i>	163
<i>Figure 10-15. Filling in a Checklist</i>	164
<i>Figure 10-16. Action Items List Page</i>	165

<i>Figure 10-17. New Action Item Page.....</i>	166
<i>Figure 10-18. Sample Email Notification for Action Items</i>	167
<i>Figure 10-19. Editing Action Item Page.....</i>	168
<i>Figure 10-20. Document Comments List Page.....</i>	169
<i>Figure 10-21. New Document Comment Page.....</i>	170
<i>Figure 10-22. Exporting a Checklist.....</i>	171
<i>Figure 11-1. Problem Report Flow.....</i>	172
<i>Figure 11-2. Select Problem Reports Page.....</i>	174
<i>Figure 11-3. Field Drop Down List Example</i>	175
<i>Figure 11-4. Problem Reports List Page</i>	176
<i>Figure 11-5. New Problem Report Page (Top Half).....</i>	177
<i>Figure 11-6. New Problem Report Page (Bottom Half)</i>	180
<i>Figure 11-7. PR Notification Email Example.....</i>	182
<i>Figure 11-8. PR Notification Email Example</i>	182
<i>Figure 11-9. Export Problem Reports Page</i>	186
<i>Figure 11-10. Import Problem Reports Page</i>	187
<i>Figure 11-11. Problem Report History List</i>	187
<i>Figure 12-1. Documents List Page</i>	189
<i>Figure 12-2. Select Documents for Certification Package</i>	190
<i>Figure 13-1. Archives Function at the Project Level.....</i>	192
<i>Figure 13-2. Archives List Page</i>	193
<i>Figure 13-3. New Archive Page.....</i>	194

1 Introduction to PACT

The *Process & Artifact Compliance Tool (PACT)* assists teams developing projects that are subject to safety-critical compliance standards. The tool is currently focused on supporting programs that need to comply with DO-178C and DO-254 for airborne software and hardware, respectively, and DO-278A for ground-based software. It may also support other safety and quality-based programs such as ISO 9001, HIPAA, TS 16949, and ISO 26262.

PACT is a second generation incarnation of the tool QCMS (Qualtech's Compliance Management System), formerly sold by Qualtech Consulting (www.FAAConsultants.com). QCMS has been retired and replaced by PACT, which includes the same functionality but with a more modern interface, some additional features, and a more maintainable, flexible, and secure infrastructure.

As with QCMS, the PACT tool is best suitable for small to medium sized companies developing projects of up to several hundred requirements.

The purpose of this document is to define the functionality of the PACT tool and guide tool users in the use of the tool.

1.1 What Does PACT Do?

PACT provides a central interface to guide your safety-critical project. The tool allows team members, which can be geographically dispersed, to see appropriate project content and interact with the data that is central to their own role. Management can easily monitor progress of the project, customers can access the data they need, and external certification authorities can review any or all data of certification concern.

PACT facilitates certification compliance with its ability to help a team:

- Enable secure, “anywhere” project access for worldwide and/or work-from-home project teams
- Store, manage, and access the many artifacts of a safety-critical project in an intuitive manner
- Hold reviews at each critical juncture in the process as well as reviews of all necessary documents and artifacts of the process, including tracking and managing Action Items and Document Comments to closure
- Provide a single repository for internal, customer & certification reviews
- Provide Ready-to-edit complete documentation sets
- Utilize Designated Engineering Representative (DER)-created and vetted review checklists
- Track and baseline documents to support configuration management

- Produce push button traceability matrices
 - Track Problem Reports in a formalized process to closure
- Download or archive a project or specific data to specified location

1.2 Understanding the PACT Graphical User Interface

This section discusses various aspects of the graphical user interface (GUI). The next section provides an overview of how to use the tool.

PACT is a web-based tool. When you first log in, you will be greeted as follows:



Figure 1-1. PACT Home Page

This is called the **PACT Home Page**. Clicking the **PACT Icon** in the top left of the page will always bring you back to the **PACT Home Page**.



Figure 1-2. PACT Home Page Icon

If you want to know the Version of PACT you are using, click on **About** in the lower right corner. Likewise, if you want to contact Airworthiness Certification Services (ACS) for any reason, including to discuss an issue about PACT, click on **Contact**.



Note that **About** and **Contact** are located in this same location on every page in PACT, and are also available under **Info** in the **Menu Bar** (see Figure 1-4) at the top of the page.

The primary use of PACT is to work on *Projects* (i.e., Systems) and their corresponding *Hardware and Software Items*. To do this, you start by clicking on **View all Projects** in the **PACT Home Page**, which brings you to this view of all the Projects that you have access to.

Identifier	Name	Navigation	Show	Edit	Delete
CPDLC	Controller Pilot Data Link	Review Status Archives Hardware/Software Items System Requirements Problem Reports Review Status Archives	Show	Edit	Delete
DME	Distance Measuring Equipment	Hardware/Software Items System Requirements Problem Reports Review Status Archives	Show	Edit	Delete
Doc	Document Example	Hardware/Software Items System Requirements Problem Reports Review Status Archives	Show	Edit	Delete
DSC	Door and Slide Controller	Hardware/Software Items System Requirements Problem Reports Review Status Archives	Show	Edit	Delete

New Project Back Undo

Process and Artifact Compliance Tool About Contact

Figure 1-3. PACT Projects List Page

On any List type of page, including the **Project List Page**, you will see column headers over pertinent information. These column headers are active links and if you click them, they will reorder the information. This can help you view/sort information in a particular way.

At the very top of the screen you will see a *Menu Bar*.

Figure 1-4. Menu Bar

This **Menu Bar** lets you to perform various actions. Clicking on **Home** or the **PACT Icon** will always return you to the **Home Page**.

Clicking **Log Out** will log you out of PACT.

Info is a dropdown menu that permits you see information about PACT (including Help) and/or setup information about yourself that may be needed for various processes. For example, you can edit your user information to upload an electronic signature image that will be needed when you sign-in to reviews.

If you are an Administrator in PACT, you will also see an **Admin** dropdown menu. This menu lets Administrators perform various special functions. For more information on the Administrator role, see Section 3.

At the very end of the **Menu Bar**, you will see another dropdown item. The name of this item varies depending on which page you are on. For the **Projects List Page** (as shown), it is called **Projects**. This menu provides access to functions for each page. For example, while most pages include a **Back** button at the bottom, this menu always provides a quick, alternative way to go **Back**. This is useful if the items on the page exceed the height of the window and you cannot see the buttons at the bottom of the page. By using this menu, you do not have to scroll to the bottom to access various functions.

Below the **Menu Bar**, in the grey area, you will see a *Breadcrumbs* bar. This indicates where you are hierarchically in the tool. In Figure 1-3, you are just down one level from the **Home Page** at the **Projects List Page** (i.e., “All Projects” in the Breadcrumbs). If you were further down in the tool hierarchy, you would see a hierarchical path to your current page. As you navigate through PACT, you can also return to previous pages by clicking a specific level (which is an active link) in the **Breadcrumbs bar**.

You may filter the objects you want to see in various List pages. For example, you can filter which projects that you want to see in the **Project List Page**. There are two methods to do this.

1. In the top left corner of any list you may choose a *Filter* selection. For example:

Field:	Identifier	Value:	<input type="button" value="Filter"/>
--------	------------	--------	---------------------------------------

Select the **Field** list item you want to search for, which corresponds to fields of information applicable to the objects shown in the List page, and then enter the **Value** for that field you want to search on. Then click the **Filter** button. The list will only display items with the specified field values.

2. In the top right corner of any list of items you will find a **Search Box**. For example:

Search:
<input type="text"/>

You may enter a value in the **Search Box** and press Enter. This will display only the items that have the specified text string in their displayed information.

The difference between the **Filter** and **Search** box choices is that **Filter** filters on data that is applicable to the objects displayed in the List even if it is not visible in the list (for example, **Access** in the **Projects**), but **Search** can only search for data that is displayed in the list.



Note that PACT has various List pages and where pertinent, these pages include the same filtering and searching capabilities, though tuned for the specific objects of the list page.

In any list of objects, you may see several icons after each one (depending on pertinent actions for and your access to those objects).



Figure 1-5. Action Buttons

These are called **Action Buttons** and they show all the actions you can perform on the object: in this case, Show (view), Edit or Delete the objects in the list.

Clicking the **Undo** () button on any screen will undo the last change that you made. For example, if you delete a Project and then decide that was not a correct decision, you can click the **Undo** button and the Project will be restored. You can click **Undo** as many times as you like. When there is nothing left to **Undo**, then the button will not be visible.

Clicking the **Redo** () button on any screen will reverse the last **Undo** operation. For example, if you **Undo** a delete to a Project and then decide that was not a correct decision, you can click the **Redo** button and the Project will be removed again. You can click **Redo** as many times as you like. When there is nothing left to **Redo**, then the button will not be visible.

Clicking the **Back** () button on any page will return you to the last that was executed or displayed. This is similar to, the Back arrow in your browser.

While editing and viewing Requirements, Source Code, Test Cases, Test Procedures and Model Files you may see **Previous** and **Next** buttons at the bottom of the page. These will take you to the previous or next Record. For example, if you are editing SYS-0002, the **Next** button will take you to SYS-0003 and **Previous** will take you to SYS-0001. If you do not see a particular button, that means that you are at the end or beginning of all the records.



While editing if you click  or  you will lose any changes you have made.

However, if you save the changes and then click  or  that will take you to the previous or next record in edit mode.

In the **Navigation** column of each Project in the Project list you will see a series of links. These links will take you to the various objects related to the Project. This is discussed further in Section 2.5.1.



Note: In various screen shots throughout this manual, you may see fields with an asterisk (). This indicates that the field is required.*

2 PACT Usage Overview

This section highlights a quick flow through the tool, showing usage at a high-level and referring to the pertinent sections that follow to get more information.

The main steps are as follows:

- 1** Set up your account
- 2** Log in
- 3** Set up your Project
- 4** Set up your Hardware/Software Items
- 5** Proceed through the Development Life Cycle for each Item (utilizing key aspects of PACT to assist with the safety-critical program requirements)

The subsections that follow summarize each of these steps.

2.1 Setting Up Your Account

After you purchase PACT, your Administrator will set up your PACT account. (See Section 3.3).

Once your account is set up, you will be provided a URL as follows:

https://acs-pact.com/pact_awc?organization=ORGANIZATION

OR

https://acs-pact-gov.com/pact_awc?organization=ORGANIZATION for ITAR supported projects.

2.2 Logging In

The first time you use PACT, you must log in. When the tool is set up, the Administrator provides a list of users who are allowed to use the tool, along with their roles (see Section 3).

As a user of PACT, you will be provided a Username and Temporary Password, along with your company's specific PACT URL.

Within the browser of your choice, type in your company's PACT URL.

This will bring you to the initial **Login Page**.

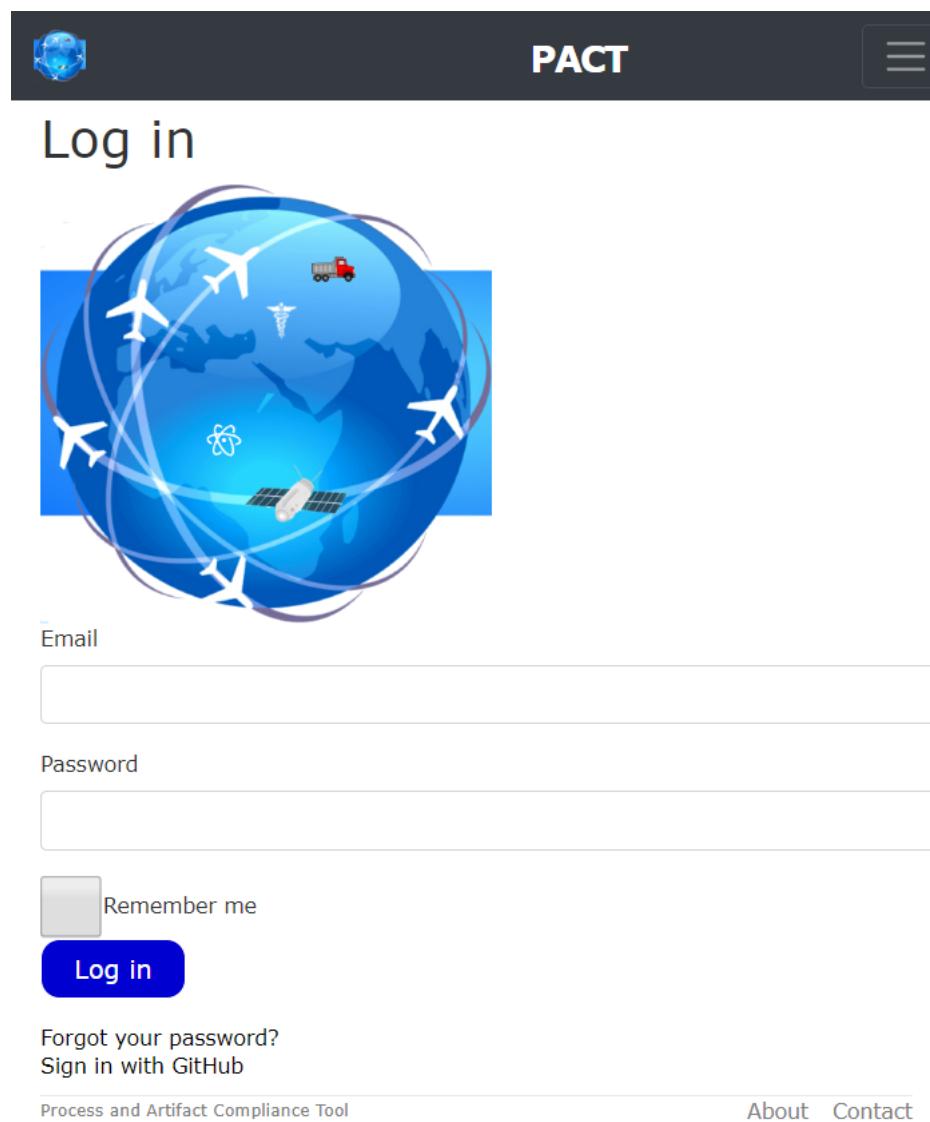


Figure 2-1. Login Page

Log in with your provided credentials and click “Remember me” if you want to skip the **Login Page** in future sessions (i.e., checking this box takes you directly to the **Home Page** next time you click on your company’s URL).

If you forgot your password and need to reset it, click **Forgot your password**. This will email your Administrator, who can issue you a password reset email.

Alternately, you **Sign in with Github** (as a Single Sign-on Solution) to use that login to get into PACT.

2.3 Changing Your Password

It is always a good idea to change your initial password. You do this by selecting **Info > Change Password** from the top **Menu Bar** of any page, as shown in Figure 2-2.

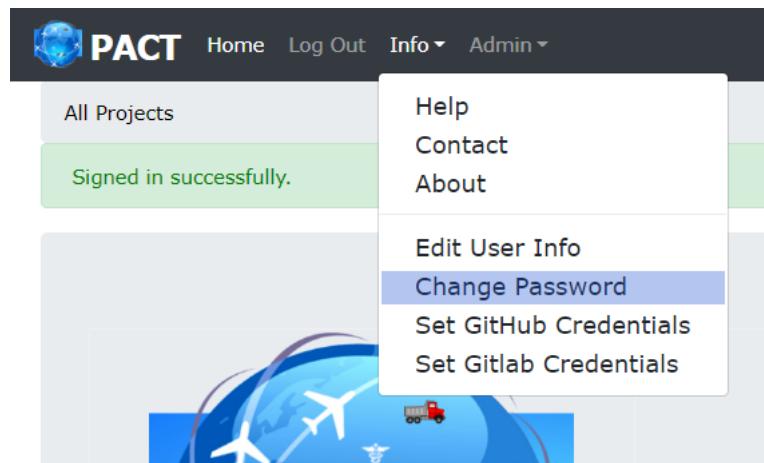


Figure 2-2. Change Password Menu Item

This brings up the **Change Your Password Page**.

A screenshot of the 'Change your password' page. The header is identical to Figure 2-2. The main content area has a heading 'Change your password'. Below it are two input fields: one for 'New password' and one for 'Confirm your new password', both marked with an asterisk. A large blue button at the bottom is labeled 'Change my password'.

Figure 2-3. Change Your Password Page

Fill in and confirm your new password. Then click the **Change my password** button.

2.4 Listing PACT Projects

When you first log into PACT, you will click on **View all Projects** from the **Home Page**. This brings you to the **Project List Page** (see Figure 1-3).

When you log in for the first time, this list may be empty. If you are the Project Manager, you will have to set up your first project.

2.5 Setting up Your First Project

Typically, it is the Project Manager or possibly the Configuration Manager who sets up a Project. Remember, in PACT, the term *Project* means the system-level project, under which are one or more hardware and/or software projects (referred to as *Items*). By establishing a Project at this higher level, you can link, reference, and review pertinent data between hardware and software items that comprise the Project.

2.5.1 Understanding a PACT Project

A PACT Project lets you do the following:

- Capture the *System Requirements* (which feed down and link to the corresponding hardware or software items)
- Navigate down into these software and hardware sub-components of the system (i.e., the *Hardware/Software Items*)
- Check the progress of these lower-level items by:
 - Monitoring their *Review Status*
 - Examining all of the *Problem Reports* from these items.
- *Archive the Project*

When you create a Project, you are defining the PACT infrastructure for the system-level design, which gives you pertinent visibility into the Hardware/Software Items that comprise that system design.

2.5.2 Creating a New Project

If your role is to create the Project, follow these steps.

- 1 On the **Home Page** (see Figure 1-1), click on **View All Projects**. This takes you to the **Projects List Page** (see Figure 1-3).
- 2 Click on the **New Project** button. This brings you to the **New Project Page**.

The screenshot shows the 'New Project' page of the PACT application. At the top, there is a navigation bar with the PACT logo, Home, Log Out, Info, Admin, and Project options. Below the navigation bar, the title 'All Projects' is displayed, followed by 'New Project'. The form fields include:

- * Identifier: A text input field.
- * Name: A text input field.
- System requirements prefix: A text input field containing 'SYS'.
- Project Managers: A list box containing 'Admin User', 'Dave Newton', and 'Michelle Lange'.
- Configuration Managers: A list box containing 'Admin User', 'Dave Newton', and 'Michelle Lange'.
- Quality Assurance Members: A list box containing 'Admin User', 'Dave Newton', and 'Michelle Lange'.
- Team Members: A list box containing 'Admin User', 'Dave Newton', and 'Michelle Lange'.
- Certification Representatives: A list box containing 'Admin User', 'Dave Newton', and 'Michelle Lange'.
- Access: A dropdown menu set to 'PUBLIC'.
- Permitted Users: A list box containing 'Admin User', 'Dave Newton', 'Michelle Lange', and 'Michelle Lange*'.

At the bottom of the page are two buttons: 'Create Project' (highlighted in blue) and 'Back'.

Figure 2-4. New Project Page

3 Fill out the fields as follows:

- **Identifier:** The **Identifier** is usually an abbreviation that identifies the Project in the context of file or requirements naming. For example, if the system is a Flight Management Computer, you may want its **Identifier** to be “FMC”. The **Identifier** is used by default in the prefixes of various files and requirements types created within a Project or Item scope.
- **Name:** This is the full name of the Project. In the example above, you would fill in “Flight Management Computer.”
- **System Requirement Prefix:** This is the abbreviation that is added to each system requirement for this project as part of the naming/numbering. By default, this prefix is “SYS”. You can override this by deleting it and typing the prefix you want in the field.
- **Project Managers:** You many have one or more *Project Managers*. Select these by clicking on a name. Once highlighted, the user is listed as the Project Manager. If you need to select more than one name, hold the SHIFT/CTRL/COMMAND/ALT key (browser dependent) while selecting one or more other names. Make sure all the names you want to be Project Managers are highlighted before moving on. (See Section 3 for more information on team member roles).
- **Configuration Managers:** In the same way as above, select one or more users to be *Configuration Managers* for the Project. (See Section 3 for more information on team member roles).
- **Quality Assurance Members:** In the same way as above, select one or more users to be *Quality Assurance Members* for the Project. (See Section 3 for more information on team member roles).
- **Team Members:** A Team Member is someone on the team who needs access to the project but does not have a specially defined role (e.g. Configuration Manager, Quality Assurance, etc.) In the same was as above, select one or more users to be *Team Members* for the Project. (See Section 3 for more information on team member roles).
- **Certification Representatives:** In the same way as above, select one or more users to be *Certification Representatives* for the Project. (See Section 3 for more information on team member roles).
- **Access:** Select whether your project is Public, Private or Protected. *Public* access permits anyone in the organization to access the project. *Private* access only permits the project to be accessed by the person who created the Project

(i.e., typically, the Project Manager). *Protected* access permits you to specify which users in the organization may access the project (even if they don't have a designated role).

- **Permitted Users:** This field only activates when "Access" is set to "Protected." Here you can select only those users you want to have access to the Project.
- 4 Once you have filled in all the above fields, you can click **Create Project** (the **Create Project** button) to create the Project. Alternately, use the **Back** button to exit without saving the New Project. You will now be placed onto a page created for that Project.

The screenshot shows the PACT Project Page for a project named "Flight Management Computer". The page has a dark header with the PACT logo and a navigation menu icon. Below the header, the breadcrumb navigation shows "All Projects / Flight Management Computer". The main content area has a blue header bar with the project name. The page is divided into sections: "Details:" and "Navigation:". The "Details:" section lists project metadata: Identifier (FMC), Project Managers (Michelle Lange), Configuration Managers (Michelle Lange), Quality Assurance (Michelle Lange*), Team Members (Admin User, Dave Newton, Michelle Lange, Paul Carrick, Paul J. Carrick*, Steve Gregor, Tammy Reeve), Certification Representatives (Tammy Reeve), and Access (PUBLIC). The "Navigation:" section lists links to Hardware/Software Items, System Requirements, Problem Reports, Review Status, and Archives. At the bottom, there are three buttons: Edit, New Project (highlighted in blue), and Back. The footer includes links to Process and Artifact Compliance Tool, About, and Contact.

Figure 2-5. Project Page

The **Project Page** provides an overview of the information you entered when you created a specific Project as well as some **Navigation** selections, as described here:

- **Hardware/Software Items:** As described already, the Hardware/Software Items are the sub-components of the system. While these can be considered projects to those teams working on them, in the PACT context we refer to them as Items. These

Hardware/Software Items are subject to their own compliance standards (e.g., DO-254, DO-178C, ...). See Section 2.6 for more information on setting these up.

- **System Requirements:** These are the requirements that define the capabilities of the System Project. These are referenced by and feed down into the Hardware/Software Items. See Section 4.2 for more information on setting these up.
- **Problem Reports:** Problem Reports are filed at the Project level, but can be filed against any Item of the Project. If you want to view **Problem Reports**, you can use the filtering/searching mechanisms, or use the buttons at the bottom of the page to view all problem reports, just those that you have filed, those assigned to you, or all those that are currently Open. See Section 11 for more information on Problem Reports.
- **Review Status:** In the Project context, this is a view only interface to the status of reviews that roll up from the lower level Hardware/Software Items that comprise the system.
- **Archives:** You can create an Archive of various aspect of the Project. See Section 13 for details.



*Note: You can get back to the **Project Page** (as shown in Figure 2-5) for a specific Project by selecting the “Show” option to the right of the specific Project you would like to view on the **Projects List Page** (as shown in Figure 1-3).*



*Tip: You can click the Right Mouse Button on any **Navigation Item** in **Project** (or other) **List Pages** and select the option of opening that Page in a new browser tab. This can be useful if you are trying to look at multiple objects in PACT, such as a Problem Report as well as the source code or other object to which the problem applies.*

2.6 Setting Up Hardware/Software Items

Once you set up a Project, the next step is to set up the supporting Hardware/Software Items.

2.6.1 Understanding a PACT Item

The hierarchy in PACT is that the top-level *Project* is the system. Below this, hardware and/or software sub-components comprise the system. While in the general sense of the term, these are developed as hardware or software projects, with dedicated project teams, in the context of PACT usage, they are called *Hardware/Software Items*.



*Note that the PACT use of the term *Item* is consistent with the definition and use of the term “Item” in the ARP 4754A (airborne systems compliance) industry document.*

Figure 2-6 shows the PACT hierarchy and the type of data contained at both the Project (blue) and Item (green) level.

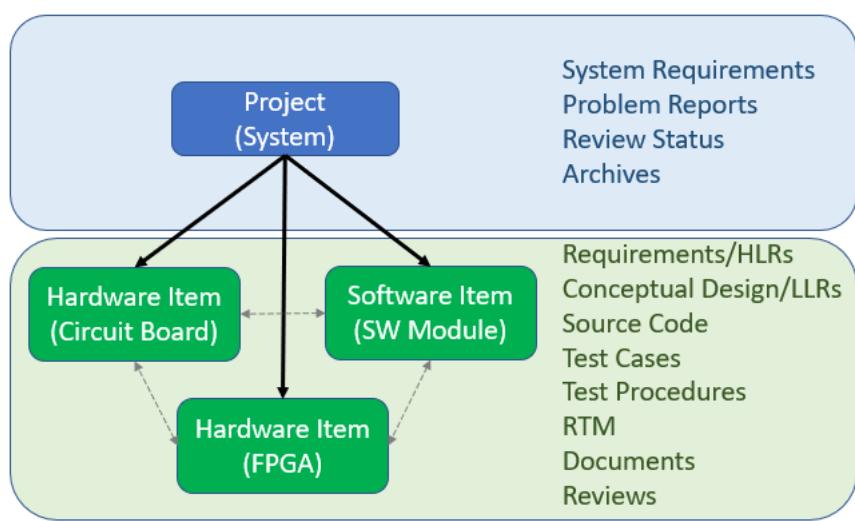


Figure 2-6. Project/Item Hierarchy in PACT

Hardware/Software Items provide a way to encapsulate and isolate logical or physical subsystems of a Project. Examples of why you might have more than one Hardware/Software Item in a Project would be that the Project contains both hardware and software components, a Project contains multiple hardware items (such as boards and multiple FPGAs), or a software project has many different modules (not to be confused with files) or components.

In order for PACT to operate properly, there must be at least one Hardware/Software Item under a Project, although you can have as many Hardware/Software Items as necessary.

Hardware/Software Items are logically separate, but it is still a good idea to ensure that objects and data below each are unique. By default, PACT names objects under each Item uniquely. This ensures at the Project level there is no duplication or confusion from information that may roll up from these Items to the Project level. You can override the object naming, but if you do so, ensure that duplication among the Item level objects is not possible.

Objects/data at the Project level (i.e., System Requirements or Problem Reports) are shared among (or roll up from) all the lower-level Hardware/Software Items.

Artifacts (i.e., artifacts) under Software Items are Documents, High-Level Requirements (HLRs), Low-Level Requirements (LLRs), Model Files, Test Cases, Test Procedures, Source Code Records, Requirements Tracing Matrices (RTMs), and Reviews.

Artifacts under Hardware Items are Documents, Requirements, Conceptual Designs, Model Files, Test Cases, Test Procedures, Source Code Records, Requirements Tracing Matrices, and Reviews.

2.6.2 Creating a New Hardware/Software Item

In the **Projects List Page** (or a specific **Project Page**) click on **Hardware/Software Items** under the **Navigation** menu. You will be taken to the **Hardware/Software Items List Page**.

The screenshot shows the PACT software interface for managing hardware and software items. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Items. Below that is a breadcrumb trail: All Projects / Flight Management Computer. The main title is "Hardware/Software Items List". On the right, there's a search bar labeled "Search:". The central part is a table with columns: Name, Identifier, Type, Level, Navigation, and three action buttons (Show, Edit, Delete). The table contains the following data:

Name	Identifier	Type	Level	Navigation	Show	Edit	Delete
A Hardware Module	AHM	DO-254 Airborne Hardware	B	Documents Requirements Conceptual Design Model Files Source Code Test Cases Test Procedures Problem Reports Requirements Tracing Matrix Reviews			
Communications Board	CommBrd_fpga	DO-254 Airborne Hardware	C	Documents Requirements Conceptual Design Model Files Source Code Test Cases Test Procedures Problem Reports Requirements Tracing Matrix Reviews			
Communications Board Software	CommSoftware	DO-178 Airborne Software	B	Documents High-Level Requirements Low-Level Requirements Model Files Source Code Test Cases Test Procedures Problem Reports Requirements Tracing Matrix Reviews			

At the bottom left are "New Item" and "Back" buttons. At the bottom right are links for "About" and "Contact".

Figure 2-7. Hardware/Software Items List Page

The first time you navigate to this page, if no **Hardware/Software Items** have yet been added, it will be empty.

The process of adding a new Hardware or Software Item is very similar to the process of setting up a Project. If it is your responsibility to set up a new Hardware or Software Item, follow these steps:

- 1 Click on **New Item** at the bottom of the **Hardware/Software Items Page**. This brings up the **New Item Page**.

The screenshot shows the 'New Hardware/Software Item' page in the PACT application. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and Item. Below the navigation bar, the current location is shown as All Projects / Flight Management Computer / Items. The main title 'New Hardware/Software Item' is displayed prominently. There are several input fields and dropdown menus for entering item details:

- * Name: A text input field.
- Type of Item: A dropdown menu.
- * Identifier: A text input field.
- Design Assurance Level: A dropdown menu.
- High-Level Requirement or Requirements Prefix: A text input field containing 'HLR'.
- Low-Level Requirements or Conceptual Design Prefix: A text input field containing 'LLR'.
- Model file prefix: A text input field containing 'MF'.
- Source code prefix: A text input field containing 'SC'.
- Test case prefix: A text input field containing 'TC'.
- Test procedure prefix: A text input field containing 'TP'.
- * Project: A dropdown menu showing 'Flight Management Computer'.

At the bottom of the form are two buttons: 'Create Hardware/Software Item' and 'Back'. Below the form, a footer bar includes links for Process and Artifact Compliance Tool, About, and Contact.

Figure 2-8. New Item Page

2 Fill out the fields in the **New Item Page** as follows:

- **Name:** This required field is the full name of the Item. For example, if you want to set up a software I/O module, you could call this “Software Input/Output”.
- **Type of Item:** You select which type of compliance this Item is subject to as follows:
 - DO-178: Airborne software
 - DO-254: Airborne hardware
 - DO-278: Ground-based software
 - DO-160: Airborne hardware subject to environmental testing

- **Other:** If your hardware or software item is not subject to one of the above standards, you can choose this option.
- **Identifier:** The Identifier, which is a required field, is usually an abbreviation that identifies the Item in the context of object (such as requirements) naming. It is usually an abbreviation of the **Name**. For example, if the Item name is “Software Input/Output”, you may want its **Identifier** to be “SW_IO”.
- **Design Assurance Level:** The **Design Assurance Level** (DAL) is the level of compliance criticality that is needed during the process. Higher DALs require a more stringent process, with more objectives that need to be met. This is A (highest) – D (lowest) for DO-178C/DO-254/DO-160 and 1-4 for DO-278. You can choose Other if your project does not need to comply with any of the listed assurance levels.
- **High-Level Requirements or Requirements Prefix:** By default, this field will be pre-filled with the Hardware/Software Item **Identifier**, followed by HLR (i.e. <ID-HLR>). This field pre-fills once you fill in the **Identifier**. Each high-level requirement itself will consist of the prefix and the next available number. You can override the default naming by typing what you’d like the Prefix of HLRs to be.
- **Low-Level Requirements or Conceptual Design Prefix:** By default, this field will be pre-filled with the Hardware/Software Identifier, followed by LLR (i.e. <ID-LLR>). Each low-level requirement itself will consist of the prefix and the next available number. You can override the default naming by typing what you’d like the Prefix to be.
- **Model File Prefix:** By default, this field will be pre-filled with the Hardware/Software Identifier, followed by Model File (i.e. <ID-MF>). Each model file ID will consist of the prefix and the next available number. You can override the default naming by typing what you’d like the Prefix to be.
- **Source Code Prefix:** By default, this field will be pre-filled with the Hardware/Software Identifier, followed by SC (i.e. <ID-SC>). Each source code ID itself will consist of the prefix and the next available number. You can override the default naming by typing what you’d like the Prefix of Source Code files to be.
- **Test Case Prefix:** By default, this field will be pre-filled with the Hardware/Software Identifier, followed by TC (i.e. <ID-TC>). Each test case object itself will consist of the prefix and the next available number. You can override the default naming by typing what you’d like the Prefix of Test Cases to be.
- **Test Procedure Prefix:** By default, this field will be pre-filled with the Hardware/Software Identifier, followed by TP (i.e. <ID-TP>). Each test procedure object itself will consist of the prefix and the next available number. You can override the default naming by typing what you’d like the Prefix of Test

Procedures to be.



Note that you can override the default numbering and default prefixes to customize them for your Project. If you choose to override any number or prefix, it is your responsibility to ensure that IDs are unique among Items.

- 3 Click **Create Hardware/Software Item** to create the new Item and to bring you to the specific **Item Page**.

The screenshot shows the PACT software interface. At the top, there is a dark header bar with the PACT logo and a navigation menu icon. Below the header, the URL 'All Projects / Flight Management Computer / Items' is visible. The main content area has a blue header bar with the text 'Hardware/Software Items: A Hardware Module'. Underneath, there are two sections: 'Details:' and 'Navigation:'. The 'Details:' section contains the following information: Identifier: AHM, Type: DO-254 Airborne Hardware, Level: B, and Project: Flight Management Computer. The 'Navigation:' section lists various links: Documents, Requirements, Conceptual Design, Model Files, Source Code, Test Cases, Test Procedures, Problem Reports, Requirements Tracing Matrix, and Reviews. At the bottom of the page are five blue buttons labeled 'Edit', 'New Item', 'Export', 'Back', and 'Undo'. Below these buttons, the text 'Process and Artifact Compliance Tool' is displayed, along with 'About' and 'Contact' links.

Figure 2-9. Specific Hardware/Software Item Page



*Note: This is the same Page you will see if you go to the **Hardware/Software Items List Page** and click "Show" for any specific Item.*

- 4 You can set up more Hardware/Software Items by following this same process.

Objects and data associated with a Hardware/Software Item can be accessed from the **Navigation** column of a specific Hardware/Software Item in the **Hardware/Software Item Page** or from the list shown on the **Hardware/Software Items List Page**. This lets you access High-Level Requirements (for Software Items) or Requirements (for Hardware Items), Low-Level Requirements (for Software Items) or Conceptual Designs (for Hardware Items), Test Cases, Test Procedures, Source Code, the Requirements Tracing Matrix, Documents and Reviews that are associated with the Item.

To understand how and when to create the remaining content (as shown under the **Navigation** menu) for each Hardware/Software Item, refer to the pertinent section elsewhere in this User Manual.

2.7 Using PACT Through the Development Life Cycle

Depending on your scope of work, as defined by your Project and its corresponding Hardware/Software Items, and the compliance standard(s) (e.g., DO-254 or DO-178C) to which you must comply, your development life cycle may vary. Regardless of the specifics of your life cycle, PACT facilitates achieving key objectives of safety-critical programs in the following ways:

- Managing Roles and Data Access: PACT defines roles and makes data access available only as needed to various team members. See Section 3 for more information on this aspect of PACT.
- Managing Requirements: PACT lets you capture, review, link, and trace requirements from System to and across Hardware/Software Items. See Section 4 for more information on this aspect of PACT.
- Supporting Documentation Management: PACT provides documentation templates for Hardware/Software Items having to comply with DO-254, DO-178C and (Coming Soon) DO-278A. By starting with templates, even though you still have to fill them in with information pertinent to your design and processes, you can save hundreds of engineering hours that would otherwise be spent on documentation set up and development. The templates also ensure that you have captured all the needed information and in doing so, reduce the risk of audit findings. Documents are also version managed to support the most stringent Configuration Management requirements of safety-critical programs. See Section 5 for more information on this aspect of PACT.
- Controlling and Reviewing Source Code: Software and/or hardware code that is developed as part of the design process can be captured, managed, and reviewed within the PACT environment. See Section 6 for more information on this aspect of PACT.

- Controlling and Reviewing Test Cases and Procedures: Test cases and procedures that are developed as part of the verification process can be captured, managed and reviewed within the PACT environment. See Section 7 for more information on this aspect of PACT.
- Incorporating and Managing Model Files: Model Files that are used as part of the development process to define functionality (i.e., requirements and/or design descriptions) of a Hardware or Software Item can be captured and managed within the PACT environment. See Section 8 for more information on this aspect of PACT.
- Generating Requirements Traceability Matrices: Requirements traceability is a key part of safety-critical programs. PACT assists by generating push-button requirements matrices that can be reviewed and/or pasted into project documentation. See Section 9 for more information on this aspect of PACT.
- Holding Reviews: Holding team reviews, covering the process itself, design milestones, and/or various documentation and data items, is an important aspect of verification and quality in a safety-critical project. Likewise, customers and certification authorities both will require periodic reviews of various aspects of the design. See Section 10 for more information on this aspect of PACT.
- Managing Problem Reports: Reviews and related verification activities will identify problems encountered with the process and/or design itself. These issues need to be captured, assigned to individuals to address them, and tracked to closure. See Section 11 for more information on this aspect of PACT.
- Producing Certification Paperwork: When Hardware/Software Item development is complete, or at potentially other times within the life cycle, various documents and data items from the Project may need to be submitted to the certification authority or overseeing customer. See Section 12 for more information on this aspect of PACT.
- Archiving Projects: Keeping the project data and development environment for future reference and possible modification is another essential element of a safety-critical project. See Section 13 for more information on this aspect of PACT.

The remainder of this User Manual details how to use PACT in support of each of these essential elements of safety-critical development life cycles.

3 Defining Team Members and Roles

Each team developing a System Project, Hardware Item or Software Item has numerous members with varying roles. For details of the team member roles in terms of their specific data access within PACT, see Appendix C: Roles & Access.

3.1 User Roles

PACT defines several different User Roles: Administrator, Project Manager, Configuration Manager, Quality Assurance and Team Member. Additionally, there are two specialized roles: View Only and Restricted View. Each role has specific access depending on the function. Roles are not exclusive; that is, one person may have many roles. These roles are described here:

- Administrator: When an Organization decides to purchase PACT, they identify an Administrator for PACT. This is the person who provides all of the information for the tool and user setup. This role also includes other special privileges. This person may or may not be directly involved with a Project and its related Hardware/Software Items. This special role in PACT is described in Section 3.3, which follows.
- Project Managers: A Project Manager typically manages the team and all activities of the development, and supports preparation and implementation of various planning, requirements, and design data for the project and/or its items.

In terms of data access, the Project Manager has full View, Create, Update, and Delete capabilities for all Project and Item data and reviews. The Project Manager can also copy templates, create new ones or modify them for the Organization.

- Configuration Managers: The Configuration Manager's role typically involves performing the following tasks:
 - Problem Reporting setup, maintenance and reporting. The CM also receives notifications when PRs change states.
 - Configuration Management setup, maintenance, and reporting
 - Revision Control to ensure that all product releases are reproducible, versioned, dated, and archived
 - Software/Hardware Configuration Index (SCI/HCI) and Environment Configuration index (SECI/HECI) preparation to ensure that a configuration index is created for the certifiable configuration
 - Data configuration for the software/hardware life cycle data generated during development and qualification
 - Tool and Release Archival performing backup and archival of all life cycle data

- Building release configurations for test and final release
- Archiving and safeguarding the production configuration
- Media recreation to ensure ability to reproduce the development and verification environments
- Independent builds of released configuration for verification testing and production
- Loading the software/hardware configuration into production environment

In terms of data access, similar to the Project Manager, the Configuration Manager has full View, Create, Update, and Delete capabilities for all Project and Item data and reviews. The Configuration Manager can also copy templates, create new ones or modify them for the Organization.

- Quality Assurance Members: The Quality Assurance role is to provide independent review of all verification activities and related objective evidence. Throughout the development life cycle, the Quality Assurance member will conduct and/or participate in all project reviews and audits and is responsible for the following verification activities:
 - Approve Software/Hardware life cycle data for transition through the development phases
 - Perform and audit verification activities per the appropriate plan.
 - Verify that no conflicts exist between plans and standards
 - Participate in high-level requirements reviews and design reviews
 - Participate in code reviews as needed
 - Conduct reviews of the outputs of the various life cycle phases
 - Conduct reviews of the test cases, procedures, and results
 - Create problem reports and validate disposition
 - Create trace matrices
 - Verify Configuration Control

In terms of data access, the Quality Assurance member has full View access to all data and can participate in reviews. In addition, they can Create and Update the following:

- Action items
- Documents/attachments
- Document comments

The Quality Assurance member also has full Edit access for:

- Requirements
 - Reviews
 - Code
 - Test Cases & Procedures
- **Team Members:** Each project has various Team Members. These typically include design and verification engineers, team leads, etc. The Team Member role is less specific than the other roles. As such, Team Member data access includes View access to all data including configuration items, reviews and checklist items. In addition, Team Members have Create and Update access to:
 - Documents (including attachments and comments)
 - Problem Reports
- Team Members have full access including Delete capabilities for:
- Requirements
 - Code
 - Test Cases & Procedures
- **Certification Representatives:** This role lets the designated Certification Representative view all Project data as part of their need to review Project compliance.
 - **View Only:** This role lets a user view all Project data but not interact with any of it. It may be suitable for an upper manager who wants to check on progress but is not directly involved in the Project.
 - **Restricted View:** This role lets a user view all specified Project data but not interact with any of it. It may be suitable for a customer who wants to check on various aspects of progress but is not directly involved in the Project.

3.2 User Role Scope

User Roles have one of two scopes: Global and Project.

Global Roles apply to functions that span the organization and cross projects. Templates, Users, and GitHub/GitLab Credentials are global roles that cross projects and so are controlled by Global Roles. For example, a Configuration Manager may need to modify the Organization's templates. Since Organizational templates are not specific to a project and cross all projects, the user's Global role controls whether or not they can edit the template. Global roles are set up in the User's configuration by the organization's Administrator.

Project roles are set up when the Project is created. For access within a Project, the access will be controlled by the project Roles. Project roles override Global roles.

3.3 Administrator Role

An Administrator is a special role assigned in PACT. Administrative functions in PACT fall under one of several categories as follows:

- User Administration: See Section 3.3.1.
- GitHub and GitLab Setup: See Section 3.3.2.
- Non-Standard or Corrective Actions: See Section 3.3.3.
- Template Management: See Section 3.3.4.

The Administrator has a special menu area enabling these functions within the upper Menu Bar.

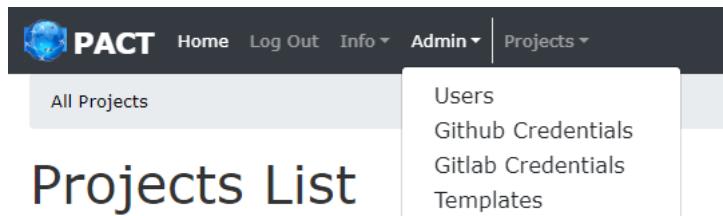


Figure 3-1. Administrator Menu

This menu is available at all times, within all Pages, during a PACT session. The following subsections describe these functions in more detail.

3.3.1 User Administration

User Administration consists of setting up new users, updating users, removing users, disabling users and forcing password resets. Part of setting up users involves setting up user roles. User roles define the access that users have to PACT functions.

This is done via the **Admin > Users** option in the Menu Bar. Clicking this opens the Users List Page. From here the Administrator can Add a New User, Edit User info or Delete Users.

The screenshot shows the 'New User' page of the PACT application. At the top, there is a navigation bar with links for Home, Log Out, Info, and Admin. Below the navigation bar, the page title 'All Projects / Users' is displayed, followed by 'New User'. The form fields include:

- Email: A text input field.
- First Name: A text input field.
- Last Name: A text input field.
- Title: A text input field.
- Phone: A text input field.
- Password: A text input field.
- Time Zone: A dropdown menu set to '(GMT-05:00) Eastern Time (US & Canada)'.
- Signature File: A file upload input field labeled 'Choose File' with 'No file chosen'.
- Profile Picture: A file upload input field labeled 'Choose File' with 'No file chosen'.
- Notify on changes: A checkbox.
- Role: A dropdown menu containing 'Project Manager', 'Configuration Management', and 'Quality Assurance'.
- Status: Two checkboxes labeled 'User disabled' and 'Password reset required'.

At the bottom of the form are two buttons: 'Create User' (in blue) and 'Back'.

Figure 3-2. New (or Edit) User Page

Adding or Editing User information here is setting the Global (i.e. Organizational) scope for the role. This can be overridden at the Project level when a Project is set up.

The New or Edit User Page includes the following fields, which the Administrator would fill in:

- **Email:** This required field specifies the User's organizational email.
- **First Name:** This required field is the User's first name.
- **Last Name:** This required field is the User's last name.
- **Title:** This is the User's organizational title.

- **Phone:** This is the User's organizational phone number.
- **Password:** This is the initial password set by the Admin. The user should be encouraged to change it when they first log in.
- **Time Zone:** This is the User's specific time zone.
- **Signature File:** This is an electronic file of the User's signature, for sign-in to checklists.
- **Profile Picture:** This is an electronic picture of the User.
- **Notify on changes:** Checking this box (RECOMMENDED) means the User will get notified by PACT via email when they are invited to a Review and/or if they have Action Items or Problem Reports assigned to them.
- **Role:** This is a required field that defines the Global role, set by the Admin. Note that this can be overridden when a Project is set up.
- **User Disabled:** Checking this box disables this User from any PACT access. This can be undone by unchecking the box, versus deleting a User, which is permanent.
- **Password Reset Required:** If checked (which the Administrator would do if they received a password reset email from a user) this will require the user to enter a new password on their next log in.

3.3.2 Github and GitLab Setup

In order to setup GitHub and GitLab access in some organizations, one must be an Administrator for GitHub or GitLab. This may not be the role of a normal PACT user. GitHub and GitLab can use certification tokens that a “normal” user might not have access to. Therefore, an Administrator is required to setup this access. This is not true of all organizations. In some cases, a “normal” PACT user could administer their own credentials. It all depends on how the organization in question set up GitHub or GitLab.

3.3.3 Non-Standard or Corrective Actions

In rare cases, some of the normal restrictions of PACT may need to be bypassed. These actions should always be approved by the CCB and/or management. The Administrator has limited ability to bypass some of PACT’s restrictions. Here are some examples:

- Deleting an In-Process Checklist: If a team member leaves the project while they are in the middle of filling in a checklist during a review. An incomplete checklist prevents a review from passing. Thus, the checklist needs to be removed. Removing a checklist cannot be done under normal roles and permissions in PACT. An Administrator must go in and remove the partial checklist.
- Renaming an Item: Normally an Item cannot be renamed once Reviews and Documents have been created for it. However, an Administrator can do this if needed.
- Changing Document Structures: If the structure of the source code tree changes in a Software Item due to refactoring, and the structure is mimicked in PACT, then the

document structure would need to be adjusted to match. The Administrator can perform this function.

3.3.4 Template Management

PACT provides Document and Checklist Templates as a starting point for documentation and reviews, respectively, to facilitate and hasten the compliance process. Based on the organization's license, certain templates will be provided to the organization by Airworthiness Certification Services (ACS). Checklist templates provide the basis for answering key questions as part of Compliance Reviews. Document templates provide the basis for compliance documentation.

In addition to the standard Templates provided by ACS, the Organization who purchases PACT may setup their own templates or customize the templates provided by ACS.



Note that at this time, Templates are only provided for DO-254 and DO-178C programs. More will be released in future tool versions.

The Administrator has a key role in setting up both types of Templates for an Organization. See Section 5.3 for information on setting up Document Templates. See Section 10.5 for information on setting up Checklist Templates.

4 Capturing and Managing Requirements

Capturing and managing requirements is a common thread among all safety-critical programs. This section describes the essential elements of requirements management and how PACT supports requirements management throughout a development life cycle.

4.1 Essential Elements of Requirements Management

Safety-critical programs are requirements-driven programs. Everything that occurs stems from a clear set of requirements, that are broken down into lower level more refined requirements. These requirements are used as the basis of development and testing. Requirements and the testing of requirements ensures that a product performs its intended function (and only its intended function).

These are the essential aspects of these programs with respect to requirements:

- Capture: Requirements are what drive the program. The first step is to define or capture the requirements. See Section 4.2.2 for how to add requirements in PACT.
- Review/Approval: Requirements should be reviewed to ensure they meet all the essential criteria. For information on how to hold a Review in PACT, see Section 10.
- Tracking Requirements Problems Changes: When problems are found in requirements, these should be formally identified in Problem Reports (see Section 11) and tracked to closure.
- Traceability: Traceability refers to the linking of requirements both throughout the requirements hierarchy (i.e., System-Level, High-Level, Low-Level) as well as to pertinent design life cycle artifacts (i.e., design, code and/or test cases/procedures). For more information on Requirements Traceability, see Section 8.

4.2 Setting Up System Requirements

Avionics and other safety critical systems mandate design flows that rely on requirements-based design. This means writing good requirements, managing them, and tracing them throughout the design and verification hierarchy.

System requirements are the starting point for any Project. These define the capabilities of the Project (i.e., system) itself. System requirements also feed down into the corresponding Hardware and Software Items that comprise the system. System requirements break down into one or more hardware or software requirements that are found within the Item scope.

You can access System Requirements by clicking the **System Requirements** link in the **Navigation** area of a specific Project. This will display the **System Requirements Lists Page**.

Requirement Number	Requirement ID	Requirement Description	Requirement Source	Safety Related?	Derived	Implementation	Model File	Version				
1	SYS001	The FMC shall have three boards: 1) communications board, 2) the processing board, and 3) the I/O board.	BOZO Aircraft Co.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Hardware		6	Show	Edit	Mark As Deleted	Delete
2	SYS002	The FMC communications board shall handle PCIe, I2C, 1553, and CAN bus communications protocols.	BOZO Aircraft Co.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Hardware and Software		3	Show	Edit	Mark As Deleted	Delete
3	SYS003	The FMC processing board shall process all Comm and I/O data.	BOZO Aircraft Co.	<input checked="" type="checkbox"/>	<input type="checkbox"/>			2	Show	Edit	Mark As Deleted	Delete
4	SYS004	this auto fill for system level requirement ID's is a bit limiting.	Steve	<input type="checkbox"/>	<input checked="" type="checkbox"/>			2	Show	Edit	Mark As Deleted	Delete
5	SYS005	another requirement. 	ICD	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Screen Shot 2020-03-31 at 2.15.15 PM.png	3	Show	Edit	Mark As Deleted	Delete

New View Allocation View Not Outlinked Export Import Renumber New Baseline View Baselines Back

About Contact

Process and Artifact Compliance Tool

Figure 4-1. System Requirements List Page

On the **System Requirements List Page**, you can do the following:

- View Details of an Existing Requirement
- Edit an existing System Requirement
- Delete or Mark a System Requirement as Deleted
- Add a new System Requirement
- View the Allocation of all the System Requirements
- View System Requirements that are not Outlinked
- Export System Requirements
- Import System Requirements
- Renumber System Requirements
- Baseline System Requirements
- View System Requirements Baselines

Each of these activities are described in the following subsections.

4.2.1 Viewing Details of an Existing System Requirement

If you have a list of existing System Requirements, you can view the details of any of them by

clicking the Show icon for a specific System Requirement or choose the Show button from the **System Requirements Lists Page**.

This brings you to the specific **System Requirement Page**, such as that shown here.

The screenshot shows a web-based application interface for managing system requirements. At the top, there's a navigation bar with a globe icon, the text 'PACT' in the center, and a menu icon on the right. Below the navigation is a breadcrumb trail: 'All Projects / Flight Management Computer / System Requirements'. The main content area has a blue header bar with the text 'System Requirement ID: SYS002'. Underneath, there's a table-like structure with rows of requirement details:

Details:	
Requirement Description:	The FMC communications board shall handle PCIe, I2C, 1553, and CAN bus communications protocols.
Requirement Category:	Comm
Requirement Verification Method:	Test
Requirement Source:	BOZO Aircraft Co.
Safety Related?	true
Implementation:	
Version:	2
Project:	Flight Management Computer

At the bottom of the page, there are several buttons: 'Edit' (white background), 'New System Requirement' (blue background), 'Back' (white background), and 'Undo' (white background). Below these buttons is a footer bar with links for 'About' and 'Contact'.

Figure 4-2. Specific System Requirement Page

The specific **System Requirement Page** shows all the information that was input during the specific requirement's creation or subsequent editing.

4.2.2 Adding a New System Requirement

When you first set up a new Project, you will not have any system requirements. They must be added. If your responsibility is to add System Requirements, you can either add them manually or Import them (see next subsection). To add System Requirements manually, perform the following steps:

- 1 On the **System Requirements List Page**, click the **New System Requirement** button. This brings up the **New System Requirement Page**.

The screenshot shows the 'New System Requirement' page in the PACT application. The page has a header with the PACT logo and navigation links for Home, Log Out, Info, Admin, and System Requirements. Below the header, the breadcrumb navigation shows All Projects / Flight Management Computer / System Requirements. The main content area is titled 'New System Requirement'. It contains several input fields: 'Requirement Number' (value: 8), 'Requirement ID' (value: SYS-0008), 'Description' (with a rich text editor toolbar containing various icons like bold, italic, underline, etc.), 'Category' (empty), 'Verification method' (dropdown menu showing 'Test', 'Review/Inspection', and 'Analysis/Simulation'), 'Source' (empty), 'Safety Related?' (checkbox), 'Derived' (checkbox), 'Derived justification' (empty), 'Implementation' (empty), 'Model File' (dropdown menu), 'Project' (dropdown menu showing 'Flight Management Computer'), and a 'Save System Requirement' button. At the bottom of the page, there are links for Process and Artifact Compliance Tool, About, and Contact.

Figure 4-3. New System Requirement Page

2 Enter the information on this page as follows:

- **Requirement Number:** This will prefill with the next available number.
- **Requirement ID:** This will prefill to the default (or overridden) system requirement prefix (that was entered when you set up the Project) followed by the next available requirement number.
- **Description:** Enter a description for this System Requirement.
- **Category:** This is a blank field that you fill in if you wish to specify the category of this system requirement. Your team, when creating the requirements, can set up the requirements categories for the project. Examples may be “video inputs”, “device specification”, “pin configuration”, or whatever makes sense for your system.
- **Verification Method:** You may select one or more verification methods that will be used to verify the design has met the requirement.



Note that in the Verification Method field, and in numerous other fields within PACT, the top selection in the drop-down is blank. This lets you

indicate that you are choosing not to specify any option.

- **Source:** You also enter the source of the requirement, in other words, who or where it came from (e.g., customer, team member, etc.).
- **Safety-Related:** If this System Requirement is safety related click the **Safety Related** checkbox.
- **Derived:** If this System Requirement is Derived click the **Derived** checkbox.
- **Derived Justification:** If you click the **Derived** checkbox you must enter the reason the System Requirement is derived in this box.
- **Implementation:** You can enter an implementation type for this system requirement. This field gives the team the flexibility to identify and use this field as they choose, which could be something like “Software” or “Hardware.”
- **Model File:** If this requirement uses a Model File you can attach it here. It will appear with the description. The Model File can be a Document already in PACT (see Section 5 on Documents) or a file directly Uploaded. To use a Document already in PACT, simply choose it. To upload a file, choose **Upload File** and then choose the file from your computer.
- **Image:** *This function is TBD.*

- 3 You can Select **Save System Requirement** to create the System Requirement or **Back** to exit without saving the System Requirement.

4.2.3 Editing an Existing System Requirement

If you have a list of existing System Requirements, you can edit them as needed.

To edit a System Requirement, from the **System Requirements Page**, click the  icon for a specific System Requirement (or choose the **Edit** button from the specific **System Requirement Page**). Editing a System Requirement is identical to creating a new System Requirement and all the fields are the same. You simply modify the content as needed and then click **Save System Requirement** at the bottom of that page.

4.2.4 Deleting or Marking a System Requirement as Deleted

You may want to Delete a requirement or simply mark it as Deleted. **Mark as Deleted** will leave the system requirement but will “strikeout” the contents. This permits you to maintain a record that the System Requirement existed at one time but was later deleted. In some cases, requirements marked as deleted will not appear. An example of this would be on an export of requirements. **Delete** will remove any trace of the System Requirement.

From the **System Requirements Page**, clicking the or icons for a specific System Requirement will delete that System Requirement. Before deleting a System Requirement, a confirmation dialog will appear asking if you are sure you want to delete the System Requirement. You must "OK" the dialog to permit the operation.

4.2.5 Viewing System Requirements Allocation

To view the System Requirements and how they are related to all the other High-Level Requirements click the button. This will bring up the **Requirements Tracing Page**.

System Requirements to Command_fpga Requirements/High-Level Requirements to Command_fpga Requirements/High-Level Requirements to I/O_Software Requirements/High-Level Requirements			
System Requirements	Command_fpga Requirements/High-Level Requirements	Command_fpga Requirements/High-Level Requirements	I/O_Software Requirements/High-Level Requirements
SYS001 The FMC shall have three boards: 1) communications board, 2) the processing board, and 3) the I/O board.	Comm_FPGA-2 another requirement.	Comm_FPGA-2 another requirement.	
SYS001 The FMC shall have three boards: 1) communications board, 2) the processing board, and 3) the I/O board.	Comm_FPGA-2 another requirement.	HLR-0001 The Comm board shall have an fpga connecting ethernet to the RF emulator.	
SYS001 The FMC shall have three boards: 1) communications board, 2) the processing board, and 3) the I/O board.	HLR-0001 The Comm board shall have an fpga connecting ethernet to the RF emulator.		
SYS001 The FMC shall have three boards: 1) communications board, 2) the processing board, and 3) the I/O board.			I/O_Software-HLR-002 the I/o board shall tx and rx 1553 data.
SYS001 The FMC shall have three boards: 1) communications board, 2) the processing board, and 3) the I/O board.			I/O_Software-HLR-003 the I/o board shall do more stuff.
SYS002 The FMC communications board shall handle PCIe, I2C, 1553, and CAN bus communications protocols.	Comm_FPGA-2 another requirement.	Comm_FPGA-2 another requirement.	
SYS002 The FMC communications board shall handle PCIe, I2C, 1553, and CAN bus communications protocols.	Comm_FPGA-2 another requirement.	HLR-0001 The Comm board shall have an fpga connecting ethernet to the RF emulator.	
SYS003 The FMC processing board shall process all Comm and I/O data.			I/O_Software-HLR-001 Test
SYS004 this auto fill for system level requirement ID's is a bit limiting.			I/O_Software-HLR-005 The I/O software shall have 2 A429 Tx channels.
SYS005 another requirement.			I/O_Software-HLR-006 The I/O software shall have 2 A429 Tx channels.
SYS005 another requirement.			I/O_Software-HLR-001 Test
SYS007			I/O_Software-HLR-006 All A429 channels shall operate at a 1gbit baud rate.
SYS007			

Figure 4-4. System Requirements Allocation (i.e. Requirements Tracing) Page

This page shows the System Requirements for the entire project in the left column. The next column over contains the High-Level Requirements associated with the System Requirement. If the Project consists of several Hardware or Software Items, there may be additional Requirement Columns, one for each separate Hardware or Software Item.

4.2.6 Viewing System Requirements that are Not OutLinked

System Requirements that have no hardware Requirements or software High-Level Requirements associated with them are referred to as Not "OutLinked". This is important to understand which System Requirements have not been implemented.

To view the System Requirements that have no associations to other Requirements click the button. This will bring up the Requirements Tracing Page.

The screenshot shows the PACT interface with the title "Requirements Tracing". Under the heading "Not Outlinked System Requirements", there are two entries:

- SYS003**: The FMC processing board shall process all Comm and I/O data.
- SYS004**: this auto fill for system level requirement ID's is a bit limiting.

At the bottom of the page, there are links for "Process and Artifact Compliance Tool", "About", and "Contact".

Figure 4-5. System Requirements Not Outlinked

4.2.7 Importing System Requirements

If you already have a set of established System Requirements that exist outside the tool, PACT lets you import them. Importing System Requirements permits you to load the System Requirements for a Project from another application, such as DOORs. You can import System Requirements from a Comma Separated Value (.csv) file or an Excel File (.xls or .xlsx).

The file should only contain data (i.e., **no separate headers or extraneous entries**) with lines in the format of:

```
id,reqid,full_id,description,source,safety,implementation,version,project_id,created_at,updated_at,organization,category,verification_method,derived,derived_justification,archive_id,soft_delete,document_id,model_file_id
```



Tip: An easy way to get any Import format is to first do an Export in CSV or XLS format and use that generated file as a template.

To import System Requirements, from the **System Requirements List Page**, follow this process:

- 1 Click the **Import** button. This brings you to the **Import System Requirements Page**.

The screenshot shows the 'Import System Requirements' page. At the top, there's a navigation bar with the PACT logo, 'Home', 'Log Out', 'Info', and 'Admin'. Below the navigation is a breadcrumb trail: 'All Projects / Process and Artifact Compliance Tool'. The main title 'Import System Requirements' is centered above a file input field labeled 'Choose File' with the placeholder 'No file chosen'. To the left of the input field is a checkbox labeled 'Duplicates permitted'. Below the input field are two buttons: 'Load System Requirements' (highlighted in blue) and 'Back'. At the bottom of the page, there's a footer with links to 'Process and Artifact Compliance Tool', 'About', and 'Contact'.

Figure 4-6. Import System Requirements Page

- 2 Choose the file you would like to import.
- 3 Click **Duplicates Permitted** if you want to permit duplicate IDs to be imported. If you do not check **Duplicates Permitted** and a System Requirement already exists with the same ID as in the file, when the file is imported it will cause an error and the file will not be imported.

4.2.8 Exporting System Requirements

Exporting System Requirements permits you to save the System Requirements for a Project to a file that can be 1) cut-and-pasted into the appropriate *<System/Hardware/Software> Requirements Document* as part of your certification documentation set stored within PACT, or 2) potentially imported into another application.

You can export System Requirements as an HTML document, PDF document, Comma Separated Value (.csv) file or an Excel File (.xls).

To export System Requirements, from the **System Requirements List Page**, follow this process:

- 1 Click the **Export** button. This brings up the **Export System Requirements Page**, like in the example figure that follows.

The screenshot shows the 'Export System Requirements' page. At the top, there's a navigation bar with the PACT logo, 'Home', 'Log Out', 'Info', and 'Admin'. Below the navigation is a breadcrumb trail: 'All Projects / Process and Artifact Compliance Tool / System Requirements'. The main title 'Export System Requirements' is centered above a dropdown menu labeled '* Export type' with the value 'HTML' selected. Below the dropdown are two buttons: 'Export' (highlighted in blue) and 'Back'. At the bottom of the page, there's a footer with links to 'Process and Artifact Compliance Tool', 'About', and 'Contact'.

Figure 4-7. Export System Requirements Page



Tip: Anytime you Export an object (like System Requirements) this opens a new tab in your browser. It is best practice to close that tab when you are finished exporting to ensure you do not end up with multiple PACT session windows open simultaneously.

- 2** Choose the type of Export you would like (HTML, PDF, CSV or XLS)
- 3** Click the **Export** button.
 - a. For HTML and PDF formats, this displays the Requirements in the newly opened Export window. You can copy and/or save this information.
 - b. For CSV and XLS formats, the file will download to your computer. The default name is the Project Name followed by “System_Requirements.”



Note that where the file is placed when you Export data in CSV or XLS formats is browser dependent. For example, in Chrome, the file is by default placed in your computer’s “Downloads” folder and may pop-up in a download bar at the screen bottom.

- 4** Close the **Export System Requirements Page**.

4.2.9 Renumbering System Requirements

Renumbering System Requirements renames all System Requirements starting a given value and incrementing by a specific value

The screenshot shows the 'Renumber' page of the PACT application. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Back. Below the navigation is a breadcrumb trail: All Projects / Requirements Tracing / System Requirements. There are three input fields with validation messages: '* Start' (empty), '* Increment' (empty), and '* Leading zeros' (containing the value '3'). At the bottom are two buttons: a blue 'Renumber' button and a white 'Back' button.

To renumber System Requirements, click **Renumber** from the **System Requirements Page**.



Renumbering requirements is something that should be done only under extreme circumstances and/or prior to the first baseline/formal review of the requirements. Otherwise, requirements history and traceability can be adversely affected.

4.2.10 Baselining System Requirements

Baselining System Requirements saves a read-only copy of the current System Requirements along with any associated Requirements. To create a new baseline, perform the steps below.

- 1 Click the **New Baseline** button. This brings up the **New System Requirements Baseline Page** as follows.

The screenshot shows the 'New System Requirements Baseline' page. The page has the following fields:

- Name:** Project: Flight Management Computer baseline of System Requirements.
- Baseline ID:** System Requirements Baseline-0.1
- Description:** Project: Flight Management Computer baseline of System Requirements.
- Pact version:** 1.7
- Revision:** (empty field)
- Draft Version:** 0.1
- Baselined At:** 07/15/2020, 10:44:07 AM

At the bottom are two buttons: **Create Requirements Baseline** and **Back**.

Figure 4-8. New System Requirements Baseline Page

- 2 Enter the information on this page as follows:

- **Name:** This is the long name used to identify the Baseline. It is prefilled but may be changed.

- **Baseline ID:** This is the short identifier used to identify the Baseline. It is prefilled but may be changed.
- **Description:** This is a description for the Baseline. It is prefilled but may be changed.
- **PACT Version:** This is version of PACT that was used to create the Baseline. It may not be changed.
- **Revision:** This is a revision number that is associated with the Baseline. It does not need to be filled in and may be left blank.
- **Draft Version:** This is the version for the Baseline. It is prefilled but may be changed. It defaults to the next available floating-point number but may be an integer or a floating-point number. A string can be entered but then the auto-incrementing will be disabled.
- **Baselined At:** This is the date and time of the baseline and may not be changed.

- 3** You can Select [Create Requirements Baseline](#) to create the System Requirements Baseline or [Back](#) to exit without saving the System Requirements Baseline. Creating the baseline may take a few minutes based on how many requirements there are.

4.2.11 Viewing System Requirements Baselines

Clicking the [View Baselines](#) button brings up the **System Requirements Baselines List Page** as follows.

Name	Identifier	Description	Revision	Draft Version	Baselined At					
Project: Flight Management Computer baseline of System Requirements.	System Requirements Baseline-0.1	Project: Flight Management Computer baseline of System Requirements.		0.1	2020-07-15 11:16AM PDT	View Contents	Show	Edit	Delete	

New Requirements Baseline Back Undo

Figure 4-9. System Requirements Baseline List Page

On the **System Requirements Baselines List Page**, you can do the following:

- View the contents of the Baseline
- View the information about the Baseline.
- Edit the information about the existing Baseline

- Delete a Baseline
- Create a New System Requirements Baseline

To view the contents of a System Requirements Baseline, from the **System Requirements**

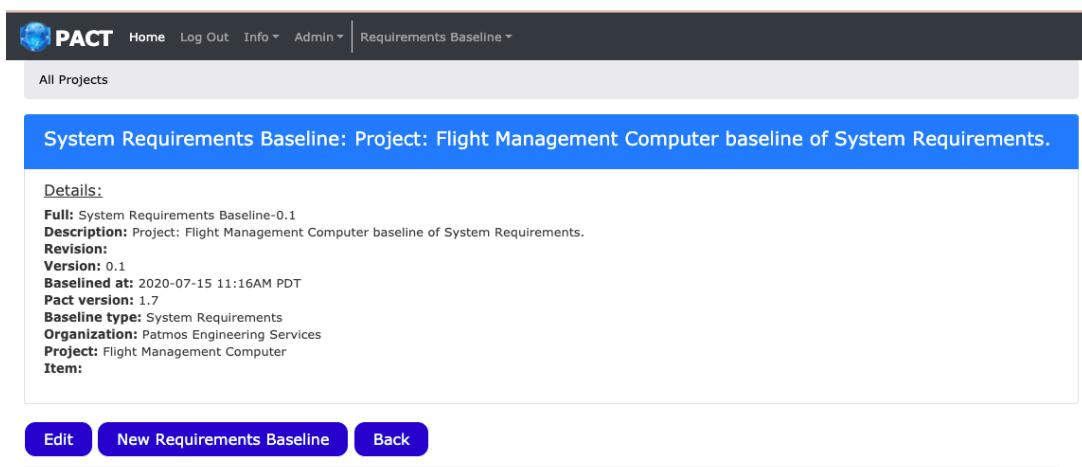
Baselines List Page, click the  icon for a specific Baseline. This will take you into the contents of the Baseline.

In the Baseline you will see a “snapshot” of the requirements at a given time. The requirements in this view cannot be changed. You will know that you are viewing a baseline by the word **Baseline** in the path on the *Breadcrumbs* bar. You can navigate in the Baseline in exactly the same manner as with all other projects. The only differences are that you are seeing the requirements as they were at a given point in time and they cannot be changed.

To leave the Baseline, return to the PACT home screen by clicking **Home** from the Menu or clicking **All Projects** in the *Breadcrumbs* bar.

To view information about a System Requirements Baseline record, from the **System**

Requirements Baselines List Page, click the  icon for a specific Baseline. This provides a Page showing all the information related to the Baseline.



The screenshot shows the PACT application interface. At the top, there is a navigation bar with the PACT logo, Home, Log Out, Info, Admin, and a dropdown for Requirements Baseline. Below the navigation bar, a breadcrumb trail shows 'All Projects' and 'System Requirements Baseline: Project: Flight Management Computer baseline of System Requirements'. The main content area has a blue header bar with the title. Below it, there is a detailed list of baseline information under the heading 'Details':
Full: System Requirements Baseline-0.1
Description: Project: Flight Management Computer baseline of System Requirements.
Revision:
Version: 0.1
Baseline at: 2020-07-15 11:16AM PDT
Pact version: 1.7
Baseline type: System Requirements
Organization: Patmos Engineering Services
Project: Flight Management Computer
Item:
At the bottom of the content area, there are three buttons: 'Edit', 'New Requirements Baseline', and 'Back'. At the very bottom of the page, there are links for 'About' and 'Contact'.

Figure 4-10. Viewing a Specific Systems Requirements Baseline

Clicking the  icon for a given Baseline lets you to edit information for that Baseline. This Page works exactly like the New System Requirement Baseline Page. Note that you can only change specific information about a Baseline and not the contents of the baseline.

From the **System Requirements Baselines Page**, clicking the  icon for a specific System Requirement Baseline will delete that System Requirement baseline. Before deleting a System Requirement Baseline, a confirmation dialog will appear asking if you are

sure you want to delete the System Requirement. You must “OK” the dialog to permit the operation.

To create an additional Baseline, you can click [New Requirements Baseline](#). The process is identical to the process in Section 4.2.10, [Baselining System Requirements](#).

4.3 Managing Hardware/Software Item Requirements

Software Items (depending on your Project requirements and certification mandates, such as development assurance level) may have two levels of requirements:

- 1) High-Level Requirements (HLRs) and
- 2) Low-Level Requirements (LLRs).

Every Software Item will have HLRs. Some will also have LLRs. How you add and manage these two levels of requirements is nearly identical. What follows is a description of the method for adding and managing requirements using HLRs as an example. You can apply the same processes to LLRs.

Hardware Items have only one level of Requirements but also have objects called Conceptual Designs. Conceptual Designs are visual or textual descriptions of how the Requirements translate into the Design itself. Hardware Requirements are the same as Software High-Level Requirements and Hardware Conceptual Designs are similar in nature to Software LLRs and are treated exactly the same in PACT.

The sections that follow describe procedures that apply to Software High- and Low-Level Requirements and Hardware Requirements and Conceptual Design. The procedures shown are for High-Level Requirements but are identical across all these “Requirement” types.

4.3.1 Listing High-Level Requirements and Requirements

Clicking **High-Level Requirements** in the **Navigation** area on **Hardware/Software Items Page** will display the list of High-Level Requirements for the Software Item.

Requirement ID	Description	Category	Safety	Robustness	Derived	System Requirements	High-Level Requirements	Verification Method	Model File	Version
I/O_Software-HLR-003	the I/O board shall do more stuff.	Requirement	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SYS001	HLR-0001			6
I/O_Software-HLR-004	the board shall never use req ID in this fashion.	Requirement	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>			Demonstration		2
I/O_Software-HLR-005	The I/O software shall have 2 A429 Tx channels.	Requirement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SYS005 SYS006	Comm_FPGA-2	Test		8
I/O_Software-HLR-006	All A429 channels shall operate at a 1ghz baud rate.	Requirement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SYS007		Test		3

New High-Level Requirement Export Import Renumber New Baseline View Baselines Back

[About](#) [Contact](#)

Figure 4-11. High-Level Requirements List Page

From here you can do the following, as described in the following sub-sections:

- Create a new High-Level Requirement
- Edit an existing High-Level Requirement
- Mark a High-Level Requirement as deleted
- Delete a High-Level Requirement
- Export High-Level Requirement
- Import High-Level Requirement
- Renumber High-Level Requirement
- Baseline High-Level Requirements
- View High-Level Requirements Baselines

4.3.2 Creating a New HLR

Clicking the **New High-Level Requirement** button from the **High-Level Requirements List Page** lets you create a new High-Level requirement.

The screenshot shows the 'New High-Level Requirement' form in the PACT application. The 'Requirement Number' field is populated with '9'. The 'Requirement ID' field contains 'I/O_Software-HLR-0009'. The 'Description' field has a rich text editor toolbar above it. The 'Category' field is empty. The 'Requirements Based Verification (Test) Method' section lists 'Test', 'Review/Inspection', and 'Analysis/Simulation'. Below this are three checkboxes: 'Safety Related?' (unchecked), 'Robustness?' (unchecked), and 'Derived?' (unchecked). The 'Derived Justification' field is empty. The 'Model File' dropdown is open. The 'Item' dropdown is set to 'I/O Board'. At the bottom, there are buttons for 'Link System Requirements', 'Link High-Level Requirements', 'Save High-Level Requirement', and 'Back'.

Figure 4-12. New High-Level Requirement Page

To add a new HLR, enter the following information:

- **Requirement Number:** This is the next available number that will be assigned to a requirement. This is not editable, with the exception that an Administrator may do so, under certain conditions.
- **Requirement ID:** This required field is prefilled with the Hardware/Software Item ID, followed by “-HLR-” followed by the next available number. This is the default and recommended naming/numbering scheme, although you can override it if your plans require a different scheme.
- **Description:** This is where you write in the actual HLR.

- **Category:** You can identify what type of category applies to this requirement. As with System Requirements, this is generally the requirements categories that your team decide to use. For instance, these might be “function,” “performance,” “interface,” and so on. The field is blank and not required, but it can be useful to sort and search on requirements of a certain category.
- **Requirements Based Verification (Test) Method:** You may select one or more methods that will be used to verify this requirement.
- **Safety Related:** Safety related requirements often require additional verification and/or feedback methods. If this HLR is safety-related, click the “Safety Related” checkbox.
- **Robustness:** If the HLR is a robustness requirement, which may entail additional verification, click the “Robustness” Checkbox.
- **Derived:** If this High-Level Requirement is Derived, which means it may not have an upward traceability link, click the “Derived” checkbox.
 - **Derived Justification:** If you click the “Derived” checkbox, you must enter the reason the HLR is derived in this box.
- **Model File:** If this requirement uses a Model File you can attach it here. It will appear with the description. The Model File can be a Document already in PACT (see Section 5 on Documents) or a file directly Uploaded. To use a Document already in PACT simple choose it. To upload a file choose Upload File and then choose the file from your computer.

You can Select [Save High-Level Requirements](#) to create the High-Level Requirement or [Back](#) to exit without saving the High-Level Requirement.

4.3.3 Viewing an HLR

Once you have one or more HLRs, you can view them by either clicking the  [Show](#) icon for a specific High-Level Requirement on the **High-Level Requirements List Page** or clicking the [Show](#) button from the **Editing High-Level Requirement Page**.

4.3.4 Editing an HLR

To edit a High-Level Requirement, from the **High-Level Requirements List Page**, click the  [Edit](#) icon for a specific High-Level Requirement or choose the [Edit](#) button from a specific **High-Level Requirement Page**.

Editing a High-Level Requirement is identical to creating a new High-Level Requirement in the sense that all the fields are the same. The only difference is the data you edit gets changed in the existing requirement.

4.3.5 Linking an HLR to a System or Item's Requirements

While creating an HLR, or when editing it, you have the option to link the HLR to one or more System Requirements and/or to another Item's High-Level Requirements. Linking an HLR to a System Requirement provides traceability to the originating System Requirement. Likewise linking an HLR to an HLR in another Item provides traceability between HLRs that are in different Items.

All HLRs should link to System Requirements (unless they are defined as *Derived Requirements*, meaning they document a design decision that does not necessarily stem directly from a higher-level requirement). Linking an HLR to another Item's HLR is a way to provide linkage to other related or defining requirements in other parts of the System. For example, a Project may have two Hardware Items and a Software Item: a board, an FPGA and a software module. HLRs from the board (for example, HLRs defining the I/O) may link to HLRs in the FPGA and software module. Providing these linkages is essential to understanding how the System requirements feed down into the Items and how the functions of these Items may be interrelated. These linkages provide the information necessary to build Requirements Traceability Matrices (RTMs).

4.3.5.1 Linking an HLR to System Requirements

To link an HLR to System Requirements, click **Link System Requirements** at the bottom of the **New** or **Edit High-Level Requirement Page**. This expands the **Link System Requirements Panel** as follows.

The screenshot shows a user interface for linking system requirements. At the top, there is a button labeled "Hide Links". Below it is a header bar with the title "Link System Requirements" and a search input field labeled "Search:". The main area contains a table with the following columns: "Requirement ID" and "Description". There are 7 rows in the table, each with a checkbox in the first column:

	Requirement ID	Description
<input type="checkbox"/>	SYS001	The FMC shall have three boards: 1) communications board, 2) the processing board, and 3) the I/O board.
<input type="checkbox"/>	SYS002	The FMC communications board shall handle PCIe, I2C, 1553, and CAN bus communications protocols.
<input type="checkbox"/>	SYS003	The FMC processing board shall process all Comm and I/O data.
<input type="checkbox"/>	SYS004	this auto fill for system level requirement ID's is a bit limiting.
<input type="checkbox"/>	SYS005	another requirement.
<input type="checkbox"/>	SYS006	Deleted
<input type="checkbox"/>	SYS007	

At the bottom of the panel, there is a button labeled "Save System Requirement Links".

Figure 4-13. Link System Requirements Panel

To link the HLR to one or more corresponding System Requirements, check the box next to the appropriate System Requirements and click **Save System Requirement Links**, which will save the specified link and hide the System Requirements Panel. You can also click **Hide Links** to hide the System Requirements Panel.

4.3.5.2 Linking an HLR to Another Item's HLRs

To link an HLR to an HLR in another Hardware or Software Item, click **Link High-Level Requirements** at the bottom of the **New or Edit High-Level Requirement Page**. This expands the **Link High-Level Requirements Panel** as follows.

The screenshot shows a user interface for linking requirements. At the top, there is a button labeled "Hide Links". Below it is a search bar with the placeholder "Choose Hardware/Software Item in which to find High-Level Requirements" and a dropdown menu containing "AHM", "CommBrd_fpga", and "CommSoftware". To the right of the search bar is a "Search:" input field. Below this is a table with columns: "Hardware/Software Item", "Requirement ID", "Description", "Safety", "Robustness", and "Derived". Two rows of data are shown:

<input type="checkbox"/>	Hardware/Software Item	Requirement ID	Description	Safety	Robustness	Derived
<input type="checkbox"/>	CommBrd_fpga	HLR-0001	The Comm board shall have an an fpga connecting ethernet to the RF hemulator.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CommBrd_fpga	Comm_FPGA-2	another requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

At the bottom left is a "Save High-Level Requirement Links" button.

Figure 4-14. Link High-Level Requirements Panel

To link the HLR to one or more corresponding High-Level Requirements, first select the Hardware/Software Item that contains the HLR you want to link to. Then check the box next to the appropriate High-Level Requirements and click **Save High-Level Requirement Links**, which will save the link and hide the High-Level Requirements Panel. You can also click **Hide Links** to hide the High-Level Requirements Panel.

4.3.6 Exporting an HLR

Exporting HLRs permits you to save the Requirements for an Item to a file that can be 1) cut-and-pasted into the appropriate *<Hardware/Software> Requirements Document* as part of your certification documentation set stored within PACT, or 2) potentially imported into another application.

You can export High-Level Requirements as an HTML document, PDF document, Comma Separated Value (.csv) file or an Excel File (.xls).

To export High-Level Requirements, from the **High-Level Requirements List Page**, click the **Export** button.

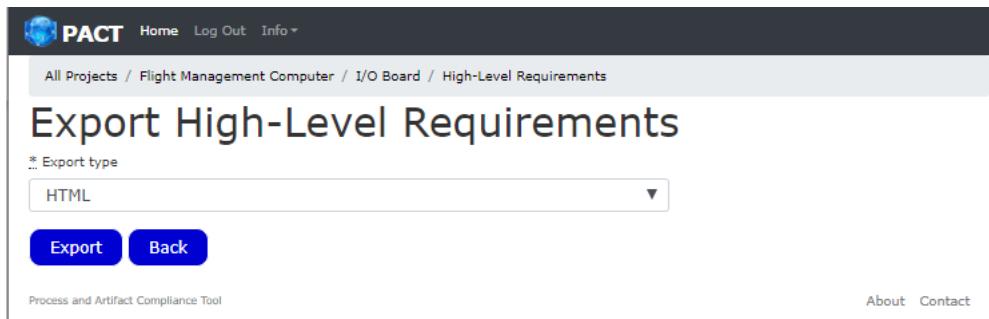


Figure 4-15. Export High-Level Requirements Page

Choose the type of export you would like (HTML, PDF, CSV or XML) and click the **Export** button.



Note that anytime you Export an object (like High-Level Requirements) this opens a new tab in your browser. It is best practice to close that tab when you are finished exporting to ensure you don't end up with multiple PACT session windows open simultaneously.



Note that where the file is placed is browser dependent. For example, in Chrome, by default the file is placed in your computer's "Downloads" folder and will likely pop-up in a download bar at the bottom of your screen.

4.3.7 Importing an HLR

Importing High-Level Requirements lets you load the High-Level Requirements for a Hardware/Software Item from another application. You can import High-Level Requirements from a Comma Separated Value (.csv) file or an Excel File (.xls).

The file should only contain data (no separate headers or extraneous entries) with lines in the format as follows (on a single line):

```
id,reqid,full_id,description,category,verification_method,safety,robustness,  
derived,testmethod,version,item_id,project_id,system_requirement_associations,  
derived_justification,created_at,updated_at,organization,archive_id,high_level_requirement_associations,soft_delete,document_id,model_file_id
```

To import High-Level Requirements, from the **High-Level Requirements List Page**, click the

Import button. This brings up the **Import High-Level Requirements Page** as follows.

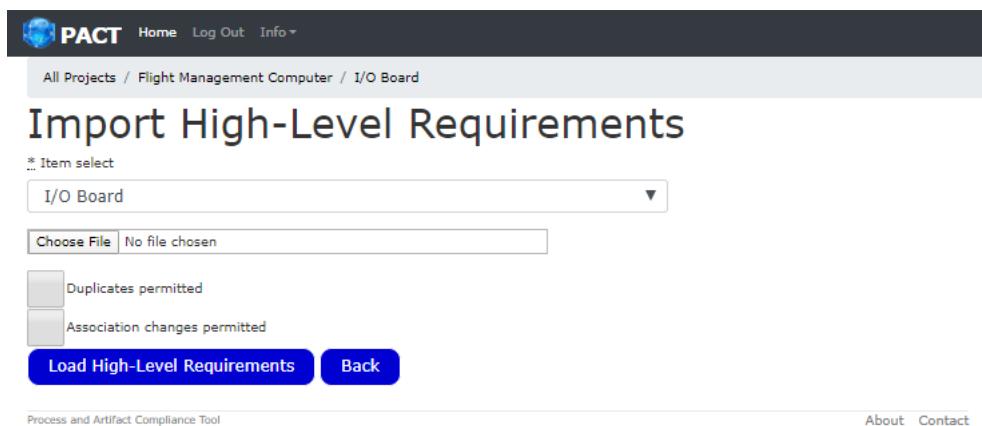


Figure 4-16. Import High-Level Requirements Page

Perform the following steps:

- 1 Choose the file you would like to import.
- 2 Select the **Duplicates Permitted** box if you want to permit duplicate IDs to be imported. If you do not check this and a High-Level Requirement already exists with the same ID as in the file when the file is imported, it will cause an error and the file will not be imported.
- 3 Click the **Association Changes Permitted** box if you want to permit changes to High-Level Requirement associations. If you do not check this and a High-Level Requirement already exists with the same ID when the file is imported and it changes the High-Level Requirement associations, it will cause an error and the file will not be imported.

4.3.8 Renumbering HLRs

Renumbering High-Level Requirements permits renumbering the High-Level Requirements starting at a given and incrementing by a specific value.

To renumber High-Level Requirements, click **Renumber** from the **High-Level Requirements** Page list.

The screenshot shows a web-based application interface for managing requirements. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Back. Below the navigation, a breadcrumb trail shows the current location: All Projects / Requirements Tracing / Software Item / High-Level Requirements. There are three input fields labeled * Start, * Increment, and * Leading zeros, each with a small dropdown arrow. Below these fields are two buttons: a blue 'Renumber' button and a blue 'Back' button. At the bottom of the page, there's a footer with links for About, Privacy and Security, and Contact. The footer also contains the company's address: PMB#224, 5500 Olympic Drive, Suite H-105, Gig Harbor Wa 98335, and phone/fax: 425-427-1956.



Renumbering requirements is something that should be done only under extreme circumstances and/or prior to the first baseline/formal review of the requirements. Otherwise, requirements history and traceability can be adversely affected.

4.3.9 Deleting HLRs or Marking HLRs as Deleted

From the **High-Level Requirements List Page**, Clicking the or icons for a specific High-Level Requirement will delete that High-Level Requirement. Before deleting a High-Level Requirement, a confirmation dialog will appear asking if you are sure you want to delete the High-Level Requirement. You must say **OK** to confirm it.

You may want to Delete an HLR or simply mark it as Deleted. **Mark as Deleted** will leave the HLR but will “strikeout” the contents. This permits you to maintain a record that the High-Level Requirement existed at one time but was deleted. **Delete** will remove any trace of the High-Level Requirement.

4.3.10 Baselining High-Level Requirements

Baselining High-Level Requirements saves a read-only copy of the current High-Level Requirements along with any associated requirements. To create a new baseline, perform the steps below.

- 1 Click the button. This brings up the **New High-Level Requirements Baseline Page**, like in the example figure that follows.

The screenshot shows a web application interface for creating a new requirements baseline. At the top, there is a navigation bar with the PACT logo and links for Home, Log Out, Info, Admin, and Requirements Baseline. Below the navigation bar, the title 'New High Level Requirements Baseline' is displayed. The form fields include:

- * Name: Project: Flight Management Computer baseline of High Level Requirements.
- * Baseline ID: High Level Requirements Baseline-0.1
- Description: Project: Flight Management Computer baseline of High Level Requirements.
- Pact version: 1.7
- Revision: (empty field)
- Draft Version: 0.1
- Baselined At: 07/16/2020, 12:18:59 PM

At the bottom of the form are two buttons: 'Create Requirements Baseline' and 'Back'.

Figure 4-17. New High-Level Requirements Baseline Page

2 Enter the information on this page as follows:

- **Name:** This is the long name used to identify the Baseline. It is prefilled but may be changed.
- **Baseline ID:** This is the short identifier used to identify the Baseline. It is prefilled but may be changed.
- **Description:** This is a description for the Baseline. It is prefilled but may be changed.
- **PACT Version:** This is version of PACT that was used to create the Baseline. It may not be changed.
- **Revision:** This is a revision number that is associated with the Baseline. It does not need to be filled in and may be left blank.
- **Draft Version:** This is the version for the Baseline. It is prefilled but may be changed. It defaults to the next available floating-point number but may be an integer or a floating-point number. A string can be entered but then the auto-incrementing will be disabled.
- **Baselined At:** This is the date and time of the baseline and may not be changed.

- 3** You can Select [Create Requirements Baseline](#) to create the High-Level Requirement Baseline or [Back](#) to exit without saving the High-Level Requirement Baseline. Creating the baseline may take a few minutes based on how many requirements there are.

4.3.11 Viewing High-Level Requirements Baselines

Clicking the [View Baselines](#) button brings up the **High-Level Requirements Baseline List Page**.

Name	Identifier	Description	Revision	Draft Version	Baselined At					
Project: Flight Management Computer baseline of High Level Requirements.	High Level Requirements Baseline-0.1	Project: Flight Management Computer baseline of High Level Requirements.		0.1	2020-07-16 12:24PM PDT	View Contents	Show	Edit	Delete	

Figure 4-18. High-Level Requirements Baselines List Page

On the **High-Level Requirements Baselines List Page**, you can do the following:

- View the contents of the Baseline
- View the information about the Baseline.
- Edit the information about existing Baseline
- Delete a Baseline
- Create a New High-Level Requirements Baseline

To view the contents of a High-Level Requirements Baseline, from the **High-Level Requirements Baselines List Page**, click the icon for a specific Baseline. This will take you into the contents of the Baseline. In the Baseline you will see a “snapshot” of the requirements at a given time. The requirements in this view cannot be changed.

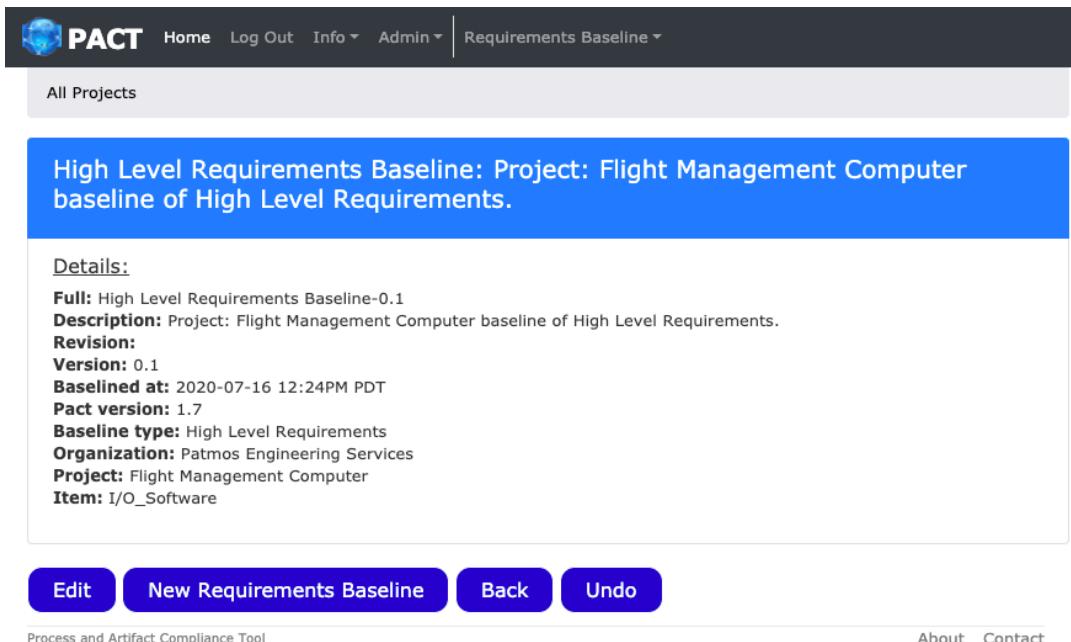
You will know that you are viewing a baseline by the word **Baseline** in the *Breadcrumbs* bar

You can navigate in the Baseline in exactly the same way as with all other projects. The only differences are that you are seeing the requirements as they were at a given point in time and you cannot change anything.

To leave the Baseline, return to the PACT home screen by clicking **Home** from the Menu or clicking **All Projects** in the *Breadcrumbs* bar.

To view a High-Level Requirements Baseline record, from the **High-Level Requirements**

Baselines List Page, click the  icon for a specific Baseline. This provides a page showing all the information related to the Baseline.



The screenshot shows a web-based application interface for PACT. At the top, there is a dark header bar with the PACT logo, a search icon, and navigation links for Home, Log Out, Info, Admin, and Requirements Baseline. Below the header, a breadcrumb trail shows 'All Projects'. The main content area has a blue header bar with the text 'High Level Requirements Baseline: Project: Flight Management Computer baseline of High Level Requirements.' In the body, there is a section titled 'Details:' containing various metadata fields with their values. At the bottom of the page are several buttons: 'Edit' (highlighted in blue), 'New Requirements Baseline', 'Back', and 'Undo'. At the very bottom of the page, there are links for 'Process and Artifact Compliance Tool', 'About', and 'Contact'.

Figure 4-19. Viewing a Specific High-Levels Requirements Baseline

 Clicking the  icon for a given Baseline will permit you to edit the information for that Baseline. This page works exactly like the **New High-Level Requirement Baseline Page**. Note that you can only change specific information about a Baseline and not the content itself.

From the **High-Level Requirements Baselines Page**, clicking the  icon for a specific High-Level Requirement Baseline will delete that High-Level Requirement baseline. Before deleting a High-Level Requirement Baseline, a confirmation dialog will appear asking if you are sure you want to delete the High-Level Requirement. You must "OK" the dialog to perform the operation.

To create an additional Baseline, you can click . The process is identical to the process in Section 4.3.10, [Baselining High-Level Requirements](#).

4.3.12 Capturing and Managing Low-Level Requirements

Clicking on **Low-Level Requirements** (or **LLRs**) from the **Navigation** area of a specific Software Item permits you to manage the Low-Level requirements of the Item.

From here you can add a new Low-Level Requirement, edit an existing Low-Level Requirement, mark a Low-Level Requirement as deleted, delete a Low-Level Requirement, export Low-Level Requirements, import Low-Level Requirements, or renumber Low-Level Requirements. Follow the same procedures as outlined in the previous sections for HLRs.

4.3.13 Capturing and Managing Conceptual Designs

As mentioned previously, Software Items use LLRs while Hardware Items use Conceptual Designs to bridge the gap between HLRs and the Design/Code itself. Within PACT, LLRs and Conceptual Designs are in essence treated the same.

Clicking on **Conceptual Designs** from the **Navigation** area of a specific Hardware Item lets you manage the Conceptual Designs of the Item.

From here you can add a new Conceptual Design, edit an existing Conceptual Design, mark a Conceptual Design as deleted, delete a Conceptual Design, export Conceptual Designs (for cut and paste into your *Hardware Design Document*), import Conceptual Designs, or renumber Conceptual Designs. Follow the same procedures as outlined previously for HLRs.

4.4 Managing Module Descriptions

The sections that follow describe procedures that apply to Module Descriptions. Module description tie source codes together into groups that are logically related. They are linked to High and Low Level requirements. In turn Source codes are linked to module descriptions.

4.4.1 Listing Module Descriptions

Clicking **Module Descriptions** in the **Navigation** area on **Hardware/Software Items Page** will display the list of Module Descriptions for the Software Item.

Number	Full ID	Description	File Names	High-Level Requirements	Low-Level Requirements	Revision	Draft Revision	Revision Date	Version					
1	MD-0001	Test Module Description	Test File Name	Software Item-HLR-1000	Software Item-LLR-1000	0.2	2			Show	Edit	Mark As Deleted	Delete	

New Module Description Export Import Renumber New Baseline View Baselines Back

Process and Artifact Compliance Tool (C)
Airworthiness Certification Services, LLC
PMB#224
5500 Olympic Drive
Suite H-105
Gig Harbor Wa 98335
Phone/fax: 425-427-1956

About Privacy and Security Contact

Figure 4-20. Module Descriptions List Page

From here you can do the following, as described in the following sub-sections:

- Create a new Module Description
- Edit an existing Module Description
- Mark a Module Description as deleted
- Delete a Module Description
- Export Module Description
- Import Module Description
- Renumber Module Descriptions
- Baseline Module Descriptions
- View Module Description Baselines

4.4.2 Creating a New Module Description

Clicking the **New Module Description** button from the **Module Descriptions List Page** lets you create a new Module Description.

This screenshot captures the 'New Module Description' page within the PACT application. The top navigation bar includes links for Home, Log Out, Info, Admin, and Module Description. The main content area is titled 'New Module Description'. It features several input fields: 'Module Description Number' (containing the value '2'), 'Module Description ID' (containing 'MD-0002'), and a large 'Description' field with a WYSIWYG editor toolbar. Below these are fields for 'Draft Version' (set to '0.1'), 'Revision' (empty), and 'Revision date' (a date input field). A dropdown menu for 'Project' is set to 'Requirements Tracing', and another for 'Item' is set to 'Software Item'. At the bottom of the form are two buttons: 'Link High-Level Requirements' and 'Link Low-Level Requirements', followed by 'Save Module Description' and 'Back' buttons.

Figure 4-21. New Module Description Requirement Page

To add a new Module Description, enter the following information:

- **Module Description Number:** This is the next available number that will be assigned to a Module Description. This is not editable, with the exception that an Administrator may do so, under certain conditions.
- **Module Description ID:** This required field is prefilled with the Hardware/Software Item ID, followed by “-HLR-” followed by the next available number. This is the default and recommended naming/numbering scheme, although you can override it if your plans require a different scheme.
- **Description:** This is where you write in the actual Module Description.
- **Draft Version:** The draft version of the Module Description
- **Revision:** The revision of the Module Description
- **Revision Date:** The date of the revision.

You can Select **Save Module Description** to create the High-Level Requirement or **Back** to exit without saving the Module Description.

4.4.3 Viewing Module Description

Once you have one or more Module Descriptions, you can view them by either clicking the  icon for a specific Module Description on the **Module Descriptions List Page** or clicking the  button from the **Editing Module Description Page**.

4.4.4 Editing a Module Description

To edit a Module Description, from the **Module Descriptions List Page**, click the  icon for a specific Module Description or choose the  button from a specific **Module Description Page**.

Editing a Module Description is identical to creating a new Module Description in the sense that all the fields are the same. The only difference is the data you edit gets changed in the existing requirement.

4.4.5 Linking an Module Description to High Level Requirements

While creating a Module Description, or when editing it, you have the option to link the Module Description to one or more High-Level Requirement. Linking a Module Description to a HLR provides traceability to the High-Level Requirement. Linking an HLR to System Requirements

To link an HLR to System Requirements, click  at the bottom of the **New** or **Edit High-Level Requirement Page**. This expands the **Link System Requirements Panel** as follows.

[Hide Links](#)

[Link Low-Level Requirements](#)

Choose Hardware/Software Item In which to find High-Level Requirements

Hardware Item	Software Item
---------------	---------------

Search:

<input type="checkbox"/> Hardware/ Software Item	Requirement ID	Description	Verification Method	Safety	Robustness	Derived
<input type="checkbox"/>	Software Item	Software Item-HLR-1000	First Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1001	First Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1002	Second Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1003	Third Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1004	Fourth Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1005	Fifth Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1006	Sixth Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1007	Seventh Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1008	Eighth Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-1009	Ninth Software High-Level SubRequirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-2000	Second Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-3000	Third Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-4000	Fourth Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-5000	Fifth Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-6000	Sixth Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-7000	Seventh Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-8000	Eighth Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Software Item	Software Item-HLR-9000	Ninth Software High-Level Requirement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

[Save High-Level Requirement Links](#)

[Save Module Description](#) [Back](#)

Figure 4-22. Link High-Level Requirements Panel

To link the Module Description to one or more corresponding High-Level Requirements, check the box next to the appropriate HLR and click [Save High-Level Requirement Links](#), which will save the specified link and hide the High-Level Requirements Panel. You can also click [Hide Links](#) to hide the High-Level Requirements Panel.

4.4.6 Linking an Module Description to Low Level Requirements

While creating a Module Description, or when editing it, you have the option to link the Module Description to one or more Low-Level Requirement. Linking a Module Description to a LLR provides traceability to the Low-Level Requirement. Linking an LLR to System Requirements

To link an LLR to System Requirements, click [Link System Requirements](#) at the bottom of the **New** or **Edit Low-Level Requirement Page**. This expands the **Link System Requirements Panel** as follows.

Hide Links

Choose Hardware/Software Item in which to find Low-Level Requirements

<input type="checkbox"/>	Hardware Item	Software Item-LLR-1000	First Software Low-Level Requirement.	<input type="checkbox"/>	Derived
<input type="checkbox"/>	Software Item	Software Item-LLR-1001	First Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-1002	Second Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-1003	Third Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-1004	Fourth Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-1005	Fifth Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-1006	Sixth Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-1007	Seventh Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-1008	Eighth Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-1009	Ninth Software Low-Level SubRequirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-2000	Second Software Low-Level Requirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLA-3000	Third Software Low-Level Requirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-4000	Fourth Software Low-Level Requirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-5000	Fifth Software Low-Level Requirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-6000	Sixth Software Low-Level Requirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-7000	Seventh Software Low-Level Requirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-8000	Eighth Software Low-Level Requirement.	<input type="checkbox"/>	
<input type="checkbox"/>	Software Item	Software Item-LLR-9000	Ninth Software Low-Level Requirement.	<input type="checkbox"/>	

Save Low-Level Requirement Links

Figure 4-23. Link Low-Level Requirements Panel

To link the Module Description to one or more corresponding Low-Level Requirements, check the box next to the appropriate HLR and click **Save Low-Level Requirement Links**, which will save the specified link and hide the Low-Level Requirements Panel. You can also click **Hide Links** to hide the Low-Level Requirements Panel.

4.4.7 Exporting an HLR

Exporting HLRs permits you to save the Requirements for an Item to a file that can be 1) cut-and-pasted into the appropriate *<Hardware/Software> Requirements Document* as part of your certification documentation set stored within PACT, or 2) potentially imported into another application.

You can export Module Descriptions as an HTML document, PDF document, Comma Separated Value (.csv) file or an Excel File (.xls).

To export Module Descriptions, from the **Module Descriptions List Page**, click the **Export** button.

The screenshot shows the PACT software interface. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, Back, All Projects, Requirements Tracing, Software Item, and Module Descriptions. The main title is "Export Module Descriptions". Below the title, there are two input fields: one for "Export type" (set to "HTML") and another for "Starting number" (empty). At the bottom of the form are two buttons: "Export" and "Back". In the footer, there is contact information for Process and Artifact Compliance Tool (C) Airworthiness Certification Services, LLC, located at PMB#224, 5500 Olympic Drive, Suite H-105, Gig Harbor Wa 98335, Phone/fax: 425-427-1956, and links for About, Privacy and Security, and Contact.

Figure 4-24. Export High-Level Requirements Page

Choose the type of export you would like (HTML, PDF, CSV or XML) and click the **Export** button.



Note that anytime you Export an object (like High-Level Requirements) this opens a new tab in your browser. It is best practice to close that tab when you are finished exporting to ensure you don't end up with multiple PACT session windows open simultaneously.



Note that where the file is placed is browser dependent. For example, in Chrome, by default the file is placed in your computer's "Downloads" folder and will likely pop-up in a download bar at the bottom of your screen.

4.4.8 Importing Module Description

Importing High-Level Requirements lets you load the module Descriptions for a Hardware/Software Item from another application. You can import High-Level Requirements from a Comma Separated Value (.csv) file or an Excel File (.xls).

The file should only contain data (no separate headers or extraneous entries) with lines in the format as follows (on a single line):

`id,module_description_number,full_id,description,file_name,version,revision,draft_revision,revision_date,high_level_requirement_associations,low_level_requirement_associations,soft_delete,project,item,archive,created_at,updated_at`

To import Module Descriptions, from the **Module Descriptions List Page**, click the

Import

button. This brings up the **Import Module Descriptions Page** as follows.

The screenshot shows the 'Import Module Descriptions' page. At the top, there's a navigation bar with the PACT logo, 'Home', 'Log Out', 'Info', and 'Admin'. Below it, a breadcrumb trail shows 'All Projects / Requirements Tracing / Software Item / Module Descriptions'. The main title 'Import Module Descriptions' is centered. Below the title, there's a dropdown menu labeled 'Item select' with 'Software Item' selected. An input field for 'Choose File' shows 'No file chosen'. There are two checkboxes: 'Duplicates permitted' (unchecked) and 'Association changes permitted' (unchecked). At the bottom, there are two buttons: 'Load Module Descriptions' (highlighted in blue) and 'Back'. At the very bottom, there's some footer text about PACT being a Process and Artifact Compliance Tool (C) by Airworthiness Certification Services, LLC, with an address at 5500 Olympic Drive, Suite H-105, Gig Harbor Wa 98335, and a phone/fax number of 425-427-1956. There are also links for 'About', 'Privacy and Security', and 'Contact'.

Figure 4-25. Import Module Definitions Requirements Page

Perform the following steps:

- 4 Choose the file you would like to import.
- 5 Select the **Duplicates Permitted** box if you want to permit duplicate IDs to be imported. If you do not check this and a High-Level Requirement already exists with the same ID as in the file when the file is imported, it will cause an error and the file will not be imported.
- 6 Click the **Association Changes Permitted** box if you want to permit changes to High-Level Requirement associations. If you do not check this and a High-Level Requirement already exists with the same ID when the file is imported and it changes the High-Level Requirement associations, it will cause an error and the file will not be imported.

4.4.9 Renumbering Module Descriptions

Renumbering Module Descriptions permits renumbering the Module Descriptions starting at one at a given value and incrementing a specific value.

All Projects / Requirements Tracing / Software Item / Module Descriptions

* Start

* Increment

* Leading zeros

3

Renumber **Back**

Process and Artifact Compliance Tool (C)
Airworthiness Certification Services, LLC
PMB#224
5500 Olympic Drive
Suite H-105
Gig Harbor Wa 98335
Phone/fax: 425-427-1956

About Privacy and Security Contact

To renumber Module Descriptionsl Requirements, **enter a starting number and an increments then** and then click **Renumber** from the **Module Descriptions Renumbner** Page.



Renumbering Module Descriptions is something that should be done only under extreme circumstances and/or prior to the first baseline/formal review of the requirements. Otherwise, requirements history and traceability can be adversely affected.

4.4.10 Deleting HLRs or Marking HLRs as Deleted

From the **High-Level Requirements List Page**, Clicking the or icons for a specific High-Level Requirement will delete that High-Level Requirement. Before deleting a High-Level Requirement, a confirmation dialog will appear asking if you are sure you want to delete the High-Level Requirement. You must say **OK** to confirm it.

You may want to Delete an HLR or simply mark it as Deleted. **Mark as Deleted** will leave the HLR but will “strikeout” the contents. This permits you to maintain a record that the High-Level Requirement existed at one time but was deleted. **Delete** will remove any trace of the High-Level Requirement.

4.4.11 Baseling High-Level Requirements

Baseling High-Level Requirements saves a read-only copy of the current High-Level Requirements along with any associated requirements. To create a new baseline, perform the steps below.

- 4 Click the **New Baseline** button. This brings up the **New High-Level Requirements Baseline Page**, like in the example figure that follows.

The screenshot shows the 'New High Level Requirements Baseline' page in the PACT application. The page has a dark header with the PACT logo and navigation links for Home, Log Out, Info, Admin, and Requirements Baseline. Below the header, a 'All Projects' section is visible. The main form is titled 'New High Level Requirements Baseline'. It contains the following fields:

- Name:** Project: Flight Management Computer baseline of High Level Requirements.
- Baseline ID:** High Level Requirements Baseline-0.1
- Description:** Project: Flight Management Computer baseline of High Level Requirements.
- Pact version:** 1.7
- Revision:** (empty field)
- Draft Version:** 0.1
- Baselined At:** 07/16/2020, 12:18:59 PM

At the bottom of the form are two buttons: 'Create Requirements Baseline' and 'Back'.

Figure 4-26. New High-Level Requirements Baseline Page

- 5 Enter the information on this page as follows:

- **Name:** This is the long name used to identify the Baseline. It is prefilled but may be changed.
- **Baseline ID:** This is the short identifier used to identify the Baseline. It is prefilled but may be changed.
- **Description:** This a description for the Baseline. It is prefilled but may be changed.
- **PACT Version:** This is version of PACT that was used to create the Baseline. It may not be changed.

- **Revision:** This is a revision number that is associated with the Baseline. It does not need to be filled in and may be left blank.
 - **Draft Version:** This is the version for the Baseline. It is prefilled but may be changed. It defaults to the next available floating-point number but may be an integer or a floating-point number. A string can be entered but then the auto-incrementing will be disabled.
 - **Baselined At:** This is the date and time of the baseline and may not be changed.
- 6** You can Select [Create Requirements Baseline](#) to create the High-Level Requirement Baseline or [Back](#) to exit without saving the High-Level Requirement Baseline. Creating the baseline may take a few minutes based on how many requirements there are.

4.4.12 Viewing High-Level Requirements Baselines

Clicking the [View Baselines](#) button brings up the **High-Level Requirements Baseline List Page**.

Name	Identifier	Description	Revision	Draft Version	Baselined At				
Project: Flight Management Computer baseline of High Level Requirements.	High Level Requirements Baseline-0.1	Project: Flight Management Computer baseline of High Level Requirements.		0.1	2020-07-16 12:24PM PDT	View Contents	Show	Edit	Delete

New Requirements Baseline Back Undo

Figure 4-27. High-Level Requirements Baselines List Page

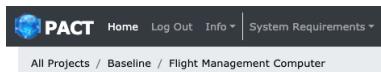
On the **High-Level Requirements Baselines List Page**, you can do the following:

- View the contents of the Baseline
- View the information about the Baseline.
- Edit the information about existing Baseline
- Delete a Baseline
- Create a New High-Level Requirements Baseline

To view the contents of a High-Level Requirements Baseline, from the **High-Level Requirements Baselines List Page**, click the  icon for a specific Baseline. This will take

you into the contents of the Baseline. In the Baseline you will see a “snapshot” of the requirements at a given time. The requirements in this view cannot be changed.

You will know that you are viewing a baseline by the word **Baseline** in the *Breadcrumbs* bar



You can navigate in the Baseline in exactly the same way as with all other projects. The only differences are that you are seeing the requirements as they were at a given point in time and you cannot change anything.

To leave the Baseline, return to the PACT home screen by clicking **Home** from the Menu or clicking **All Projects** in the *Breadcrumbs* bar.

To view a High-Level Requirements Baseline record, from the **High-Level Requirements**

Baselines List Page, click the icon for a specific Baseline. This provides a page showing all the information related to the Baseline.

The screenshot shows a web-based application interface for PACT. At the top, there is a navigation bar with the PACT logo, Home, Log Out, Info, Admin, and Requirements Baseline. Below the navigation bar, a breadcrumb trail shows All Projects / Baseline / Flight Management Computer. The main content area has a blue header bar with the text "High Level Requirements Baseline: Project: Flight Management Computer baseline of High Level Requirements". Underneath this, there is a detailed list of baseline information, including:

- Full:** High Level Requirements Baseline-0.1
- Description:** Project: Flight Management Computer baseline of High Level Requirements.
- Revision:**
- Version:** 0.1
- Baselined at:** 2020-07-16 12:24PM PDT
- Pact version:** 1.7
- Baseline type:** High Level Requirements
- Organization:** Patmos Engineering Services
- Project:** Flight Management Computer
- Item:** I/O_Software

At the bottom of the content area, there are several buttons: Edit, New Requirements Baseline, Back, and Undo. At the very bottom of the page, there are links for Process and Artifact Compliance Tool, About, and Contact.

Figure 4-28. Viewing a Specific High-Levels Requirements Baseline

Clicking the icon for a given Baseline will permit you to edit the information for that Baseline. This page works exactly like the **New High-Level Requirement Baseline Page**. Note that you can only change specific information about a Baseline and not the content itself.

From the **High-Level Requirements Baselines Page**, clicking the  icon for a specific High-Level Requirement Baseline will delete that High-Level Requirement baseline. Before deleting a High-Level Requirement Baseline, a confirmation dialog will appear asking if you are sure you want to delete the High-Level Requirement. You must “OK” the dialog to perform the operation.

To create an additional Baseline, you can click . The process is identical to the process in Section 4.3.10, [Baselining High-Level Requirements](#).

4.5 Requirements Traceability

Requirements traceability is a key part of any safety-critical program. Section 4.3.5 described how you can link new requirements to existing requirements. This can be done for both hardware and software requirements at all levels. This is how requirements traceability is established and when you link requirements while they are being created, this lets you establish traceability during development. If you do this, generating traceability matrices becomes a push-button process. See Section 9 for more information on Requirements Traceability.

5 Managing Documentation

Every safety-critical project, such as those for DO-254 and DO-178C, has stringent documentation requirements to ensure critical aspects of both the development process and design itself are captured. The documentation set for such a project starts with planning and standards documents, which must be approved and followed throughout the life cycle, and proceeds through the development life cycle with documentation capturing all the key outputs from each phase of the process. Documentation, including review and proper control of documents, is a central tenant of safety-critical programs and is a key feature supported by PACT.

This section describes a typical documentation flow within a development life cycle, and how to get started using the PACT documentation features.

5.1 Documentation Throughout a Life Cycle

Figure 5-1 shows a very basic flow for how documentation is managed in a safety-critical project. The steps that follow describe how you manage this documentation process with PACT.

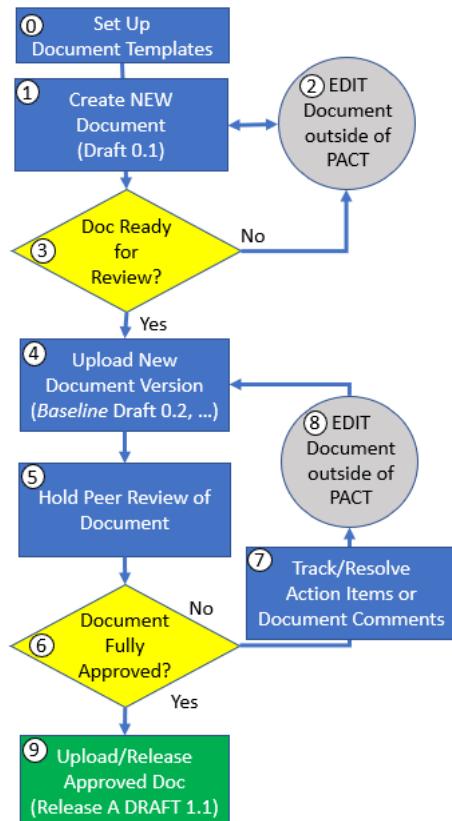


Figure 5-1. Documentation Flow

Before you start the Document flow, you should set up your Organization's Document Templates. See Section 5.3 for more information.

Then follow the steps for Document creation and management as follows:

- 1 Create:** Create a New Document in PACT. This informs PACT as to the document's existence and the tool will now start controlling the versions for your project. The first version will be Draft Version 0.1, and it will increment with each new revised upload. You can add new Documents yourself or they can come from instantiated templates (which occurs when a Hardware or Software Item is created).
- 2 Edit:** Perform the necessary editing on that document in an environment outside of PACT (such as Microsoft Word).
- 3 Review Readiness:** Determine if the document is ready for a formal *Peer Review* (which will be conducted within the framework of PACT). If it is not ready, continue editing. When it is ready, go to Step 4.
- 4 Upload:** If it is ready for a formal Peer Review, upload the revised document. The *Draft Version* will increment to 0.2 (or whatever is the next 0.X number if you uploaded earlier revisions). This Draft Version will now be considered the *Baseline* for the review.
- 5 Review:** Hold a Peer Review of the baselined document. (See Section 10 for information on how to hold Reviews in PACT).
- 6 Document Issues/Approval:** If the Peer Review found issues with the document, go to step 7 (i.e., track/resolve issues and more document editing). If the document is ready for approval, go to step 8.
- 7 Track/Resolve Issues:** Issues discovered in the Peer Review should be captured as either Action Items (see Section 10.9.1) or Document Comments (see Section 10.9.2). These issues should be resolved (typically by step 8, editing the document) so the review can pass.
- 8 Edit.** When the document issues have been addressed go back and repeat Steps 4, 5, and 6, until Step 6 is Yes.
- 9 Approve/Release Document:** Once the document is deemed complete in terms of all review (checklist) items being appropriately addressed (i.e., passing), the document can be approved. At this point, you would do another upload of the reviewed/approved document and this time indicate that it is a formal Release. PACT will then call this Release A with a Draft Version of 1.0.



Note that if later in the life cycle the team finds issues that tie back to the document, iterate through the process again starting at Step 7. Also note that a second formal Release would be labeled B. For more information on PACT Versioning see Section 5.2.

5.2 Understanding PACT Versioning

Before you start adding, editing and reviewing documents, it is important to understand the PACT scheme used to track versions.

In PACT, Documents are versioned. Each time a Document is updated, older versions are archived and a History is kept. As each document is updated, the Draft Version is incremented. For a first Draft, the version will start with 0.1 and increments to 0.2, 0.3, etc. with each new uploaded version. When you are ready for a review, you will hold a Peer Review (See Section 10) on the document. The version you review will be the document Baseline.

Most likely, you will find items that are wrong with the document and the team will need to go back and make changes. You will then upload the new version of the document (whose Draft Version will now increment to the next 0.X number and will be the new Baseline) and hold another review.

You may iterate through this cycle numerous times until all Review issues are addressed in the document and the document is considered *Approved*. At this point you are ready for an official *Release*.

When you are ready to release a document, you then specify a Release in PACT. This changes the Version to include an upper case alphanumeric sequential letter starting with A (for the first Release), then B (for the second Release), then C and so on.



Note: PACT does not use letters "I , Q, O, S, X, Z" to avoid confusion.

The first approved version of the document will then be Revision A with Version 1.0. If, after approving (i.e., releasing) the document, you find issues with the document that require changes, you will upload a new version (or versions) with these changes. These will increment so each successive draft revision will be 1.1, 1.2, 1.3, and so on.

When you review and approve these changes, you will Release the document again and it will now be Revision B, and it will be initially numbered 2.0 with each successive draft revision being incremented to 2.1, 2.2, 2.3, and so on.

5.3 Document Templates

This section describes what Templates are and how they are used to facilitate documentation (and reviews) in PACT.

5.3.1 What are Templates?

Templates are documents (or checklists) that will be used over and over again across numerous Items. This avoids having to recreate documents (or checklists) every time you start a new Project or Item. Templates can be copies of existing templates that you change or new templates you have created. Templates become the first draft revision of a Document when an Item is created. In other words, all of the available templates (pertinent to the Item) become Documents and will appear in the Documents List for each hardware/software Item created.

If you are a team member who needs to use Documents but does not need to worry about the templates from which these came, skip to Section 875.4 to learn more about Documents.

5.3.2 Document Templates Scope

PACT Templates have one of two scopes: ACS or Organizational. There are also two types of templates: Document and Checklist. The templates become available within the project, specifically in the context of Hardware/Software Items. The appropriate documents are instantiated into an Item's document list when the Item is set up. The appropriate checklists are added to a review when reviews for the Item are set up.

The templates start with the general set of ACS (Master) Templates. ACS Templates are set up and maintained by ACS. Access to ACS Templates are controlled by the Organization's license with ACS. If the license allows it, the ACS Templates are available to the Organization for use.

Organizational Templates are maintained by the Organization and are used across that Organization's PACT Projects.

Figure 5-2 shows the relationship of ACS Templates provided as part of PACT to the Organization Templates for use in the Organization's Projects.

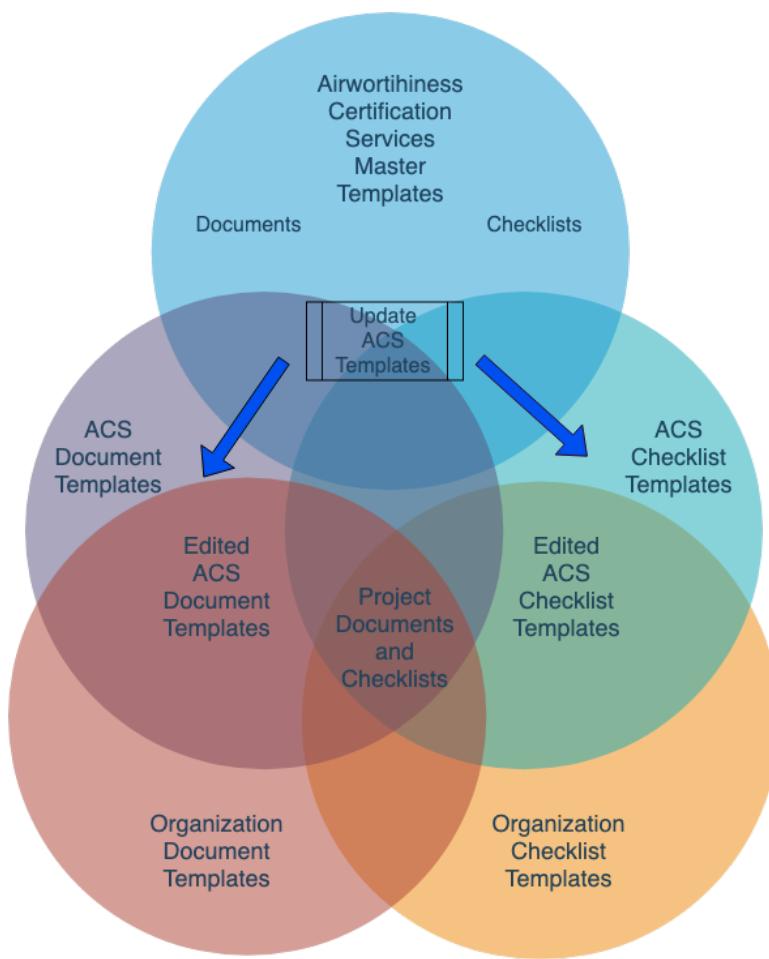


Figure 5-2. Template Scope and Flow

5.3.3 Setting Up and Using Document Templates

Setting up Document Templates is usually a job for the Project Manager or Configuration Manager, in conjunction with the Organization's PACT Administrator.

Depending on which set(s) your organization purchased, PACT is pre-loaded with a set of ACS document templates that support DO-254 and DO-178C programs. (Coming soon – DO-278A!). These can automatically be utilized by the Administrator to set up a Template Set for your company.

To ensure the latest ACS templates are in use, the Administrator can update the organization's copy of the ACS Templates. To avoid overwriting a customer's modified templates when ACS updates their templates, a copy of the ACS Master Templates is kept for each organization under the ACS Document/Checklists Templates.

To update to the latest ACS templates, the Admin can use the **Admin > Update ACS Templates** option on the Menu Bar as follows.

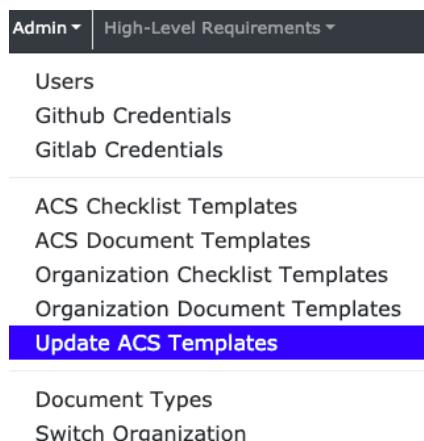


Figure 5-3. Administrator Access to Templates

This may take a few minutes. The **Templates List Page** will appear when the operation is done.

The screenshot shows the 'Templates List' page. At the top, there is a navigation bar with the PACT logo, 'Home', 'Log Out', 'Info', 'Admin', and 'Templates'. Below the navigation is a search bar labeled 'Search:' with a placeholder 'All Projects'. The main area is titled 'Templates List' and contains a table with one row. The table has columns for 'Title', 'Description', 'Template Type', 'Template Class', and 'Notes'. The first row contains the text 'Organization Template'. To the right of the table are three icons: a magnifying glass for 'Show', a pencil for 'Edit', and a trash can for 'Delete'. At the bottom of the table are buttons for 'New Template', 'Set Default Templates', 'Back', and 'Undo'. At the very bottom of the page, there are links for 'Process and Artifact Compliance Tool', 'About', and 'Contact'.

Figure 5-4. Templates List Page

As mentioned, there are two scopes in PACT. One is the copy of the Airworthiness Certification Services (ACS) templates copied in when the Administrator chose **Update ACS Templates**. The other is the Organization's own specific templates or copies of ACS template that the organization has modified.

An organization may want to alter a template from ACS. Once edited, the template is copied into the organization's templates. This permits the Administrator to update the templates from ACS without destroying changes that an organization has made.

An organization may also choose to create their own templates (see Section 5.3.4 that follows). These go into the Organization Templates as well.

When a new Item is created in a Project in PACT, all the appropriate templates are copied from both the ACS and the Organization's Templates. The templates will be copied into the Item based on the Item type (DO-178, DO-254, etc.) and DAL. These copies of the documents and checklists can be directly edited in the Item. They are no longer templates but are part of the data used in the Project.

Changes to the Documents and Checklists in the Item will not change any Documents used in future projects. In addition, changes to the Organization's Templates will not overwrite the documents and checklists in the Item.

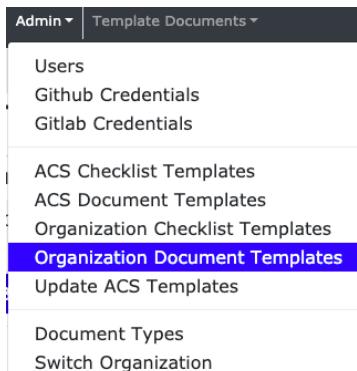
5.3.4 Creating New Templates

You can create a template in several ways: from an existing Document, from an existing Template, or from a Document outside of PACT.

5.3.4.1 Creating a New Template from an Existing PACT Document

To setup a New Template from a document that exists already within PACT, follow these steps:

- 1 Go to the **Documents List Page**, find the Document you want to use as a starting point for that template.
- 2 Click the **Download** Icon to the right of the Document you want to start with for your template.
- 3 Open the downloaded document and save it to the name and location you want for your template.
- 4 Edit the Document as appropriate to turn it into an organizational template.
- 5 If you are not the Admin, you will have to have an Admin perform the remaining steps. If you are the Admin, from the menu choose **Admin > Organization Document Templates**.



The **Organization Template Documents List Page** will appear.

All Projects

Organization Template Documents List

Template Document ID	Title	Description	Document Type	Document Class	Notes	File
864	PSAC-A Plan for Software Aspects of Certification	Template	DO-178			

New Template Document Back Undo

Show Edit Delete

Process and Artifact Compliance Tool About Contact

Figure . Documents List Page

- 6 Click **New Template Document**. This brings you to the **New Template Document Page**.

All Projects

New Template Document

* Template Document ID
865

* Template Title

Description

Source

Document type
Template

Document class

DAL

Control Category

Document ID

Document Name

Document Revision

Document Draft Version
0.0

Choose File No file chosen

Download Current File:

Notes

Create Template Document New Template Document Back Undo

Process and Artifact Compliance Tool About Contact

Figure 5-7. New Template Document Page

7 Enter the information on this page as follows:

- **Template Title:** This is the title used to Identify the Template. It may or may not be related to the file name for the document. For example, you may want to ensure the word “Template” is in the title to ensure that team members know this is a template when it appears in a Documents List.
- **Description:** This a long description for the Template, including whatever information is pertinent to its use.
- **Source:** The source for the template (i.e., who or where it was created). This might be ACS or the organization or an external source.
- **Document Type:** This is prefilled with “Template” but may be changed. For example, you might want to call it something more specific, such as a “PHAC Template for PCB Design.”
- **Document Class:** This is type of Item to which this template applies, e.g., DO-178, DO-254, etc.
- **DAL Level:** This is the DAL for the Item to which this template applies. For example, if a template should only become a document for a DAL A item, select DAL A.



NOTE: If a template applies to more than one DAL, you must go through the process of adding a new template for each DAL to which it applies.

- **Control Category:** The Control Category that this template applies to, e.g., CC1/HC1 or CC2/HC2.
- **Document ID:** This is the Document ID (i.e., abbreviated document name) that shows up when this template’s document is copied into an Item.
- **Document Name:** This is the longer, more descriptive Document Name that shows up when this template’s document is copied into an Item
- **Document Revision:** This will be the starting Document Revision that shows up when this template’s document is copied into an Item’s Document List.
- **Document Draft Revision:** This will be the starting Draft Version that shows up when this template’s document is copied into an Item’s Document List.
- **Choose File:** This is where you select the document to upload as the template.
- **Notes:** Fill this in if there are any notes related to the template.

8 Click [Create Template Document](#).

Now when you view the **Organization Template Documents List Page**, you will see the newly-created Template listed with the name and other information you provided.

5.3.4.2 Creating a New Template from Your Own Document

This process uses steps 5-8 as described in the previous section. The main difference is in Step 7, navigate to and choose the file that you have created outside of PACT to be the document template.

5.3.4.3 Creating a New Template from an ACS Template

Create a new template by editing an ACS template as follows:

- 1 Start by choosing **Admin > ACS Document Templates** from the menu.

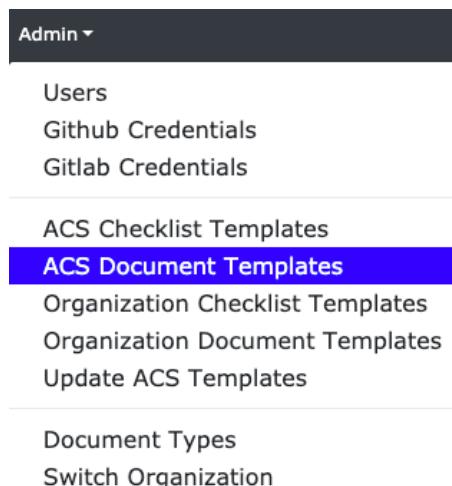


Figure 5-5. Accessing ACS Document Templates

- 2 The **ACS Template Document List Page** will appear.

ACS Template Documents List

Template Document ID	Title	Description	Document Type	Document Class	Notes	File	Show	Edit	Delete	Download
1	HPAP-A	Hardware Process Assurance Plan	Template	DO-254	Imported from /app/app/templates/do-254/documents/05_HPAP.POSTED.r3-IN-PROGRESS.doc.	05_HPAP.POSTED.r3-IN-PROGRESS.doc	Show	Edit	Delete	Download
1	SQAP-A	Software Quality Assurance Plan	Template	DO-178	Imported from /app/app/templates/do-178/documents/S_05_C_SQAP.DOWNLOAD.r2.doc.	S_05_C_SQAP.DOWNLOAD.r2.doc	Show	Edit	Delete	Download
2	SQAP-B	Software Quality Assurance Plan	Template	DO-178	Imported from /app/app/templates/do-178/documents/S_05_C_SQAP.DOWNLOAD.r2.doc.	S_05_C_SQAP.DOWNLOAD.r2.doc	Show	Edit	Delete	Download
2	HPAP-B	Hardware Process Assurance Plan	Template	DO-254	Imported from /app/app/templates/do-254/documents/05_HPAP.POSTED.r3-IN-PROGRESS.doc.	05_HPAP.POSTED.r3-IN-PROGRESS.doc	Show	Edit	Delete	Download
3	SQAP-C	Software Quality Assurance Plan	Template	DO-178	Imported from /app/app/templates/do-178/documents/S_05_C_SQAP.DOWNLOAD.r2.doc.	S_05_C_SQAP.DOWNLOAD.r2.doc	Show	Edit	Delete	Download
3	HDD-A	Hardware Design Document	Template	DO-254	Imported from /app/app/templates/do-254/documents/10_HDD.POSTED.r2.doc.	10_HDD.POSTED.r2.doc	Show	Edit	Delete	Download
4	HDD-B	Hardware Design Document	Template	DO-254	Imported from /app/app/templates/do-254/documents/10_HDD.POSTED.r2.doc.	10_HDD.POSTED.r2.doc	Show	Edit	Delete	Download
4	SQAP-D	Software Quality Assurance Plan	Template	DO-178	Imported from /app/app/templates/do-178/documents/S_05_C_SQAP.DOWNLOAD.r2.doc.	S_05_C_SQAP.DOWNLOAD.r2.doc	Show	Edit	Delete	Download
5	SWRD-A	Software Requirements Document	Template	DO-178	Imported from /app/app/templates/do-178/documents/S_09_C_SWRD.DOWNLOAD.r2.doc.	S_09_C_SWRD.DOWNLOAD.r2.doc	Show	Edit	Delete	Download

Figure 5-6. ACS Template Documents List Page

- 3** Click the  icon next to the Template you want to modify. This brings up the **Editing Template Document Page**.

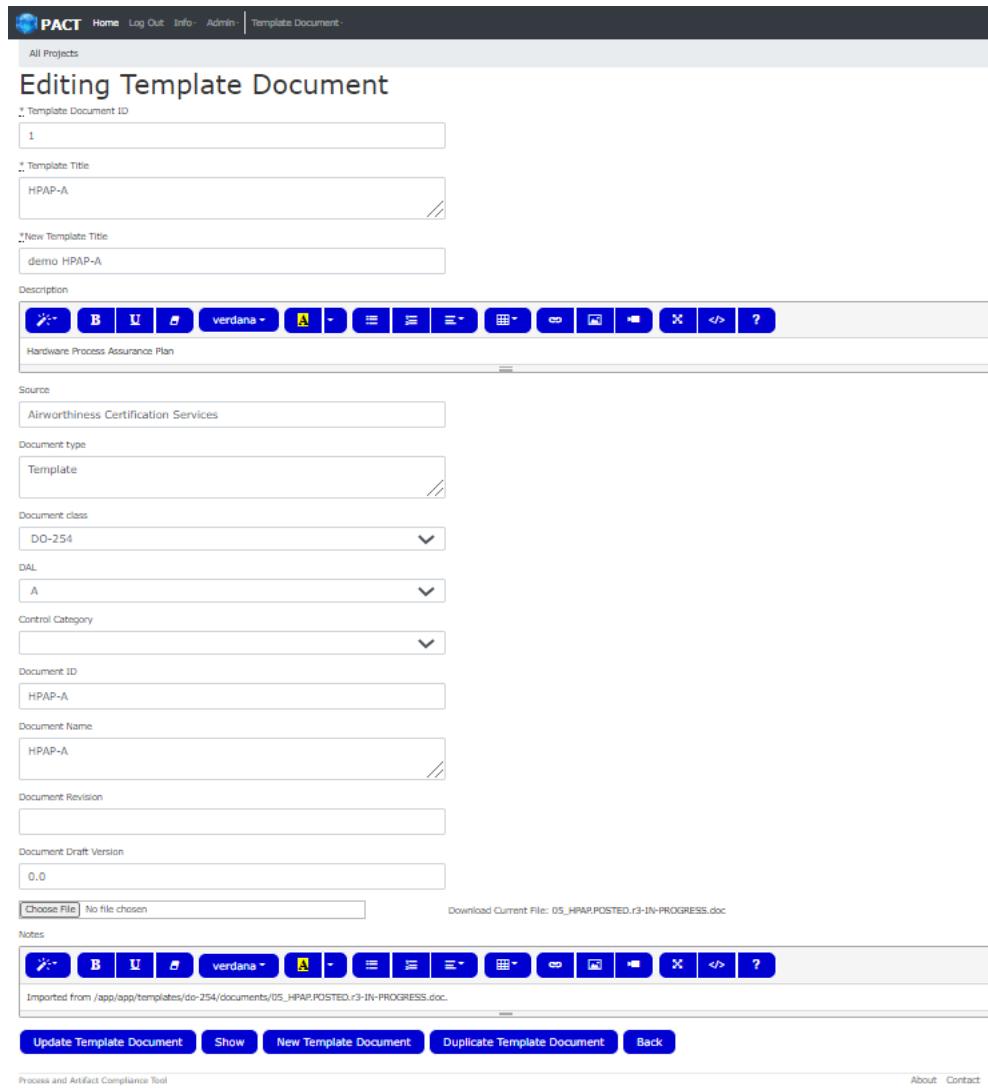


Figure 5-7. Editing Template Document Page

- 4** This page is identical to the **New Template Document Page** with one exception. To the right of the **Choose File** field button is a link that says **Download Current File**. Click the filename to download the file.
- 5** Open the file and edit it as desired.
- 6** Click the **Choose File** Button, and then choose the file you edited.
- 7** Click **Update Template Document**. The **Organization Template Document List** Page will appear showing that the Template was copied into the Organization Template Documents area with the existing name of the Template preceded by the name of the Organization.

Template Document ID	Title	Description	Document Type	Document Class	Notes	File	Show	Edit	Delete	Download
1	iee SRS-A	Software Requirements Standards	Template	DO-178		S_06_C_SRS.DOWNLOAD.r2.doc				
1	iee PSAC-A	Plan for Software Aspects of Certification	Template	DO-178		S_01_C_PSAC.DOWNLOAD.r2.doc				

New Template Document Back Undo

Process and Artifact Compliance Tool About Contact

Figure 5-8. Organization Template Documents List Page

The Templates shown in Organization Template Documents List, along with those from the original ACS Templates, will be instantiated as Documents and placed into an Item's Document List (see next subsection) when a hardware/software Item is created. The appropriate team members will have to determine which documents are pertinent to the Item and delete those that are not. The remaining subsections in this Section are related to managing the Documents themselves.

5.4 Listing the Documents

Documents are associated with a Hardware/Software Item. In other words, when you develop an Item to DO-254, DO-178C or some other standards, this will require the creation of an accompanying document set that captures nearly every aspect of the development of that Item. These documents include plans, standards, design and verification documents, and a summary of the project. The document set for safety-critical programs can be quite extensive.

To see the list of documents associated with a Hardware/Software Item, click the **Documents** link in the **Navigation** column for a specific Hardware/Software Item. This will display a **Documents List**, which lists all the Documents associated with that Item.

Documents List

Document ID	Name	Control Category	Revision	Revision Date	(Open/Closed) Comments	Show	Delete	Upload	Download
S_22_VV_SW_Evidence.DOWNLOAD.r1.xls	S_22_VV_SW_Evidence.DOWNLOAD.r1.xls				(0/0) Comments	Show	Delete	Upload	Download
S_01_C_PSAc.DOWNLOAD.r2.doc	S_01_C_PSAc.DOWNLOAD.r2.doc	Planning Document	b	2020-03-09	(3/0) Comments	Show	Delete	Upload	Download
S_02_C_SDP.DOWNLOAD.r2.doc	S_02_C_SDP.DOWNLOAD.r2.doc	CC1/HCI	E	2020-04-02	(5/0) Comments	Show	Delete	Upload	Download
S_03_C_SVP.DOWNLOAD.r3.doc	S_03_C_SVP.DOWNLOAD.r3.doc				(0/0) Comments	Show	Delete	Upload	Download
S_04_C_SCMP.DOWNLOAD.r2.doc	S_04_C_SCMP.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_05_C_SQLP.DOWNLOAD.r2.doc	S_05_C_SQLP.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_06_C_SRS.DOWNLOAD.r2.doc	S_06_C_SRS.DOWNLOAD.r2.doc	CC2/HC2	B	2020-03-26	(0/0) Comments	Show	Delete	Upload	Download
S_07_C_SD5.DOWNLOAD.r2.doc	S_07_C_SD5.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_08_C_SCS.DOWNLOAD.r2.doc	S_08_C_SCS.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_09_C_SRQ.DOWNLOAD.r2.doc	S_09_C_SRQ.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_10_C_SD0.DOWNLOAD.r2.doc	S_10_C_SD0.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_13_SVCP.DOWNLOAD.r1.doc	S_13_SVCP.DOWNLOAD.r1.doc				(0/0) Comments	Show	Delete	Upload	Download
S_14_C_SCAR.DOWNLOAD.r2.doc	S_14_C_SCAR.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_15_C_SECI.DOWNLOAD.r2.doc	S_15_C_SECI.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_16_C_SCI.DOWNLOAD.r2.doc	S_16_C_SCI.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_20_C_SAS.DOWNLOAD.r2.doc	S_20_C_SAS.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_21_CIA_DOWNLOAD.r1.doc	S_21_CIA_DOWNLOAD.r1.doc				(0/0) Comments	Show	Delete	Upload	Download
S_23_SCR.DOWNLOAD.r1.doc	S_23_SCR.DOWNLOAD.r1.doc				(0/0) Comments	Show	Delete	Upload	Download
Archived Documents	Archived Documents				(0/0) Comments	Show	Delete	Upload	Download
Instrumented Code	Instrumented Code				(0/0) Comments	Show	Delete	Upload	Download

New Document Package Certification Documents Back Undo

About Contact

Figure 5-9. Documents List Page

From here you can add a new Document, upload a Document, delete a Document, or Download a document. You can also create a certification package of documents (see Section 12).

There are two special folders that may appear in the **Documents List**. “Instrumented Code” contains source code generated by PACT when it automatically instruments the code for using the Code Coverage feature. See Section 6.12 for more on this topic. “Archived Documents” contains older versions of Documents when a document is changed. This can help you trace document history and changes from one draft version to another if needed.

5.5 Viewing a Document

To view information about a Document, from the **Documents List Page**, click the  Show icon for a specific Document or choose the  Show button from the specific **Document Page**.

5.6 Downloading a Document

In the **Documents List Page** clicking on the  (Download) icon will download or open the current version of the Document (based on your browser configuration), unless the Document is a folder (indicated by a folder icon ) in which case it will open the folder.

This is useful, for example, if you are starting with the Templates. You would download the document and edit it with your project data in your own environment to create your first version of the Item's Document. See next section for more details.

5.7 Creating a New Document

Clicking the  button from the **Documents List** lets you to create a new Document.

The screenshot shows the 'New Document' form in the PACT application. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Document. Below that, a breadcrumb trail shows 'All Projects / Flight Management Computer / Hardware or Software Item / I/O Board'. The main title 'New Document' is centered above a series of input fields. The fields include:

- * Document ID: A text input field.
- Name: A text input field.
- Control Category: A dropdown menu.
- Document type: A dropdown menu.
- Choose File: A button with 'No file chosen' text.
- Review: A dropdown menu.
- Review status: A dropdown menu.
- Release document: A checkbox.
- Draft Version: A text input field containing '0.1'.
- Revision: A text input field.
- Revision Date: A text input field with a placeholder 'mm/dd/yyyy'.
- Folder: A dropdown menu.
- * Item: A dropdown menu currently set to 'I/O Board'.

At the bottom of the form are two buttons: 'Create Document' and 'Back'.

Figure 5-10. New Document Page

To create a new Document, fill in the Fields as follows:

- **Document ID:** Enter the Document ID, which is an abbreviated name and/or number scheme for the document.
- **Name:** Enter a full name for Document. This is required.
- **Document Type:** You should select a Document Type from the pre-defined types (e.g., Plan for Software Aspects of Certification) or Other. Unless the Document Type is "Folder" you must attach a file.
- **Control Category:** This field is intended to refer to the Control Category for the document (such as CC1, HC2, etc.). If you pick a Document Type that is pre-defined in PACT, then the control category is automatically set based on the Item Type (i.e., DO-254 or DO-178C) and DAL, although you can override this if you choose.

- **Choose File:** If you are starting with a new document, here you navigate to and select the document you want to add. If you are editing a document and want to point to a newer version, this is where you select that document.
- **Review:** Documents are reviewed during Peer Reviews, and if a reviewed is set up for this document, the field will prefill with this information. This field generally applies only when you Edit a document, not add a new one.
- **Review Status:** If a review is set up for this document, the status will be shown. This field generally applies only when you Edit a document, not add a new one.
- **Release Document:** This field is greyed out for a New Document. If you are editing a document and you are ready to indicate it is a formal Release (i.e., it has been reviewed and approved), click this button.



Warning: If you check the Release Document box you cannot go back to earlier versions. Checking this creates a Release of the Document that is a CC1 controlled item under Problem reporting. A Released Document is expected to have a Closed (approved) Review. In other words, once a document is reviewed and approved by the team of assigned evaluators, it may be Released and is strictly controlled from that point on.

- **Draft Version:** This will be prefilled by the tool as per the versioning information described in Section 5.2.
- **Revision:** This will be prefilled by the tool as per the versioning information described in Section 5.2.
- **Revision Date:** This will be prefilled by the tool.

You can Select [Create Document](#) to create the Document or [Back](#) to exit without saving the Document.

5.8 Uploading a Document

When you need to change information about a document and/or bring a new version of a



document into PACT, click the [Upload](#) icon.

The **Upload Document Page** has all the same fields as the **New Document Page**.

5.9 Deleting Documents



From the **Documents List**, Clicking the [Mark As Deleted](#) or [Delete](#) icons for a specific Document will delete that Document. Before deleting a Document, a confirmation dialog will appear asking if you are sure you want to delete the Document. Select OK to delete or Cancel.

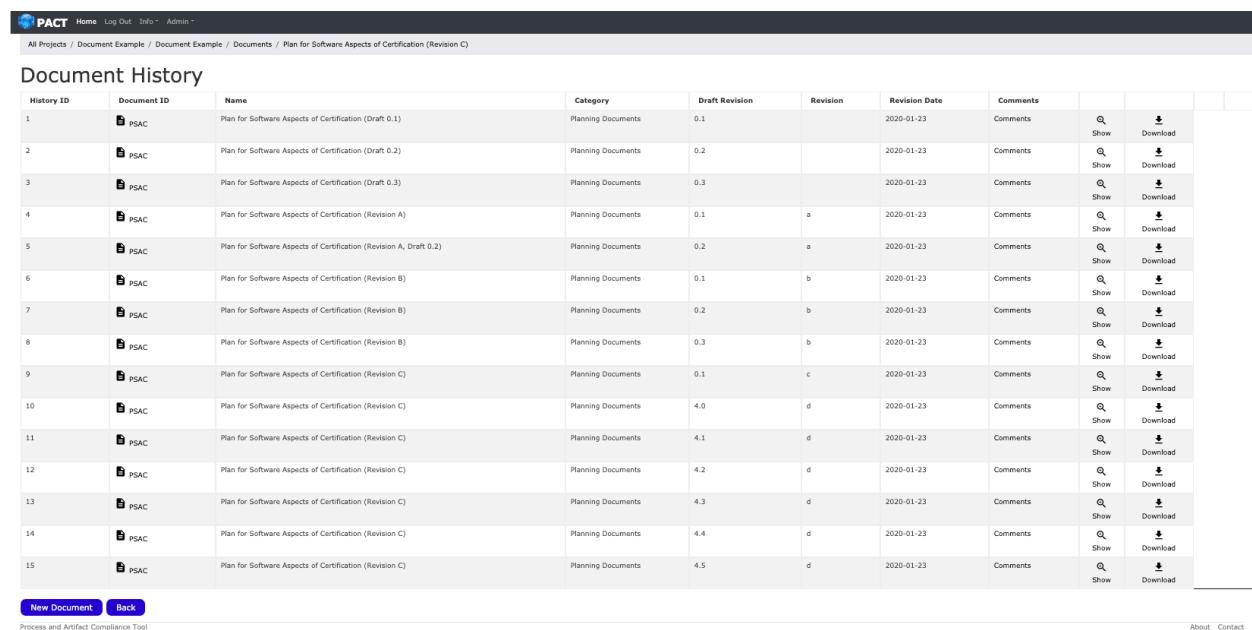


Note: You cannot delete a Document that has been formally Released (i.e., it is Revision A or greater) or that is attached to a Review.

5.10 Viewing Document History

In order to see the history of a specific document, view information about the document in the

Documents List by clicking the  icon for a specific Document or choosing the  button from the **Document Edit Page**. Then click the  button.



The screenshot shows a 'Document History' table with the following data:

History ID	Document ID	Name	Category	Draft Revision	Revision	Revision Date	Comments	Show	Download
1	PSAC	Plan for Software Aspects of Certification (Draft 0.1)	Planning Documents	0.1		2020-01-23	Comments		
2	PSAC	Plan for Software Aspects of Certification (Draft 0.2)	Planning Documents	0.2		2020-01-23	Comments		
3	PSAC	Plan for Software Aspects of Certification (Draft 0.3)	Planning Documents	0.3		2020-01-23	Comments		
4	PSAC	Plan for Software Aspects of Certification (Revision A)	Planning Documents	0.1	a	2020-01-23	Comments		
5	PSAC	Plan for Software Aspects of Certification (Revision A, Draft 0.2)	Planning Documents	0.2	a	2020-01-23	Comments		
6	PSAC	Plan for Software Aspects of Certification (Revision B)	Planning Documents	0.1	b	2020-01-23	Comments		
7	PSAC	Plan for Software Aspects of Certification (Revision B)	Planning Documents	0.2	b	2020-01-23	Comments		
8	PSAC	Plan for Software Aspects of Certification (Revision B)	Planning Documents	0.3	b	2020-01-23	Comments		
9	PSAC	Plan for Software Aspects of Certification (Revision C)	Planning Documents	0.1	c	2020-01-23	Comments		
10	PSAC	Plan for Software Aspects of Certification (Revision C)	Planning Documents	4.0	d	2020-01-23	Comments		
11	PSAC	Plan for Software Aspects of Certification (Revision C)	Planning Documents	4.1	d	2020-01-23	Comments		
12	PSAC	Plan for Software Aspects of Certification (Revision C)	Planning Documents	4.2	d	2020-01-23	Comments		
13	PSAC	Plan for Software Aspects of Certification (Revision C)	Planning Documents	4.3	d	2020-01-23	Comments		
14	PSAC	Plan for Software Aspects of Certification (Revision C)	Planning Documents	4.4	d	2020-01-23	Comments		
15	PSAC	Plan for Software Aspects of Certification (Revision C)	Planning Documents	4.5	d	2020-01-23	Comments		

Figure 5-11. Document History Page

5.11 Commenting on a Document

PACT offers the concept of commenting on a document. This is generally used in the context of reviewing a document (see Section 10). The commenting feature lets team members capture comments indicating that aspects of a document need to be changed. The comments do not get inserted into the document but instead are captured within PACT, tied to a version of the document, and assigned to a Team Member to resolve.

You can access the document comment feature in two ways: 1) from the **Documents List Page** (see Figure 5-9), and 2) from within a Checklist associated within a Peer Review.

To access Document Comments from the **Documents List Page**, look at the column with the heading (**Open/Closed Comments**).

Revision Date	(Open/Closed) Comments	
	(0/0) Comments	Show
2020-03-26	(5/0) Comments	Show
	(0/0) Comments	

Figure 5-12. Document Comments

The text in this column is an active link and it indicates the number of open/closed comments that are associated with a Document. If you click it, this brings up the **Document Comments List Page**.

Comment ID	Requested By	Assigned To	Status	Document Revision	Comment			
1	Michelle Lange	Steve Gregor	Open		Figure 1-1 has an issue.	Show	Edit	Delete
2	Michelle Lange	Michelle Lange	Closed		Figure 2 is mis-named. ML: 4/9/20: FIXED	Show	Edit	Delete
3	Michelle Lange	Steve Gregor	Open		Page 3 has a typo. DO-178B should be DO-178C.	Show	Edit	Delete

New Document Comment Go to Document

Process and Artifact Compliance Tool About Contact

Figure 5-13. Document Comments List Page

Here you can use the icons on the right to Show, Edit or Delete comments.

Click on **New Document Comment** to add a new Document Comment. This brings up the **New Document Comment Page**.

The screenshot shows the 'New Document Comment' page. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Document Comment. Below the navigation is a header 'All Projects' and a title 'New Document Comment'. A blue banner at the top indicates the document is 'Document: S_20_C_SAS.DOWNLOAD.r2.doc'. The main form contains several input fields: 'Commentid' (value: 3), 'Opened By' (value: Michelle Lange), 'Assigned To' (value: Steve Gregor), 'Status' (value: Open). There's also a 'Document Revision' field which is empty. Below these fields is a 'Comment' section containing a rich text editor toolbar with icons for bold, italic, underline, strikethrough, superscript, font family (verdana), font size (13), alignment (center, right, justify), and various other styling options like lists and tables. A message in the comment area says 'Page 3 has a typo. DO-178B should be DO-178C'. At the bottom of the page are buttons for 'Create Document comment' and 'Back', and links for 'Process and Artifact Compliance Tool', 'About', and 'Contact'.

Figure 5-14. New Document Comment Page

Editing a Comment is like capturing a simplified Problem Report (see Section 11) to track issues in the document. The Document Comment starts in the Open status. You fill in the appropriate information, assign a Team Member to address the issue, and then Create it. Once the Document Comment has been resolved, the Team Member who opened it should Edit it (by

clicking  from the **Documents Comments List Page**) and set its status to Close.

To access Document Comments from within a Review, see Section 10.9.2.

5.12 Reviewing a Document

Documents are reviewed within Peer Review processes. To understand how documents are integrated into the Review process, see Section 10.8.

6 Managing Source Code

Within a Hardware/Software Item, you can manage the Source Code shown in the Source Code List.

Clicking the Source Code link in the **Navigation** Column of a specific Hardware/Software Item will display the **Source Code List Page**.

The screenshot shows the PACT Source Code List page. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and Source Code. Below the navigation bar, the URL is All Projects / Flight Management Computer / Test. The main title is "Source Code List". There are search fields for "Field:" and "Value:" with a "Find" button, and a "Search:" input field with a magnifying glass icon.

Source Code ID	Description	Filename	URL	Functions	Module Descriptions	Git Version	Version				
1	Standard Make File	https://gitlab.faconsultants.com/patmos/demo-project/blob/master/libhash/Makfile	https://gitlab.faconsultants.com/patmos/demo-project/blob/master/libhash/Makfile.html	clean install uninstall		bc6b76092267b55c3539ec0c620f67e092569c72	1	Show	Edit	Mark As Deleted	Delete
2	nmake makefile	https://gitlab.faconsultants.com/patmos/demo-project/blob/master/libhash/Makefile.nmake	https://gitlab.faconsultants.com/patmos/demo-project/blob/master/libhash/Makefile.nmake.html	clean		bc6b76092267b55c3539ec0c620f67e092569c72	1	Show	Edit	Mark As Deleted	Delete
3	READ ME file.	https://gitlab.faconsultants.com/patmos/demo-project/blob/master/libhash/README.md	https://gitlab.faconsultants.com/patmos/demo-project/blob/master/libhash/README.html			bc6b76092267b55c3539ec0c620f67e092569c72	1	Show	Edit	Mark As Deleted	Delete
4	Lib Hash Primary C File	https://gitlab.faconsultants.com/patmos/demo-project/blob/master/libhash/libhash.c	https://gitlab.faconsultants.com/patmos/demo-project/blob/master/libhash/libhash.c.html	<pre>static uint32_t hash_dobbs(const char *key, size_t len) static uint32_t hash_murmur(const char *key, size_t len) static uint32_t hash_city(const char *key, size_t len) static uint32_t hash_spooky(const char *key, size_t len) uint32_t hash(const char *key, size_t len) static struct hash_item *hash_lookup(struct hash *, const char *key, uint32_t *hash) static void hash_set(struct hash *, const char *key, uint32_t hash) static void hash_get(struct hash *, const char *key, uint32_t *hash) void hash_destroy(struct hash *) void hash_set_destroy(struct hash *, void (*destroy)(void *)) void *hash_get(struct hash *, const char *key) int hash_set(struct hash *, const char *key, void *val) int hash_del(struct hash *, const char *key) void *hash_get_and_del(struct hash *, const char *key)</pre>		bc6b76092267b55c3539ec0c620f67e092569c72	1	Show	Edit	Mark As Deleted	Delete

At the bottom of the page, there are buttons for New, Export, Import, Scan GitHub, Scan GitLab, Renumber, Analyze, View Analysis, Set Up Coverage, Process, Profile, New Baseline, View Baselines, Zip Source Code, Back, and Undo. There is also a footer with copyright information for Airworthiness Certification Services, LLC, and links for About, Privacy and Security, and Contact.

Figure 6-1. Source Code List Page

This displays a list of Source Code *records*. The term “record” is used here because what you are doing is notifying PACT of the existence of the Source Code and providing information about it (see full definition in Appendix B: Glossary of Terms). The record may be pointing to an external URL or a document actually loaded into PACT. The List contains information about the Source Code through its record.

From here you can add a new Source Code record, edit existing Source Code, mark a Source Code as deleted, delete a Source Code record, export Source Code, import Source Code, scan Source Code from GitHub, scan Source Code from GitLab, or renumber Source Code records. These operations are explained in the following subsections.

6.1 Creating a New Source Code Record

From the **Source Code List Page**, clicking the **New** button lets you to create a new Source Code record.

The screenshot shows the 'New Source Code' form in the PACT application. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Source Code. Below that is a breadcrumb trail: All Projects / Flight Management Computer / Test / Source Code. The main form has several sections:

- * Source Code Number:** A text input field containing "34".
- * Source Code ID:** A text input field containing "Test-SC-0034".
- Description:** A rich text editor toolbar with buttons for bold, italic, underline, strikethrough, font (verdana), size (13), alignment, lists, tables, and other document controls. A green 'G' icon is visible in the top right corner of this section.
- * Filenames:** Another rich text editor toolbar with similar controls to the description section.
- GitLab Version:** A text input field.
- File Type:** A dropdown menu showing "No Attachment".
- Module:** A text input field.
- Functions:** A rich text editor toolbar with controls for bold, italic, underline, strikethrough, font (verdana), size (13), alignment, lists, tables, and other document controls.
- * Item:** A dropdown menu showing "Test".
- Buttons at the bottom:** Save Source Code, Back, Undo, Link High-Level Requirements, and Link Low-Level Requirements.

Figure 6-2. New Source Code Page

Here are the fields you use to create a New Source Code record in PACT.

- **Source Code ID:** This is prefilled by the tool by the default (i.e., <Item ID>-SC>) or user defined source code prefix by the next available number.
- **Description:** This is a description of the source code.
- **File Names:** Enter the file names of the files that contains this Source Code.
- **GitLab Version:** If the file is in an external repository, enter the version (SHA) for these files.
- **File Type:** File Type lets you attach the source code file, in one of three ways: **External URL**, **PACT Document**, or **File Upload**. Alternately you can specify **No Attachment**.
 - If you want to use an external URL (like from GitHub choose External URL). A new **URL** field will appear below the **File Type** field. Paste or type the URL in the field and that will attach the source code via the external URL.
 - On the other hand, if you already added the source code files in PACT, you can choose **PACT Documents**. This causes PACT File field to appear where you select the file from the list of Documents within PACT.
 - If you would rather not use an External URL or first upload the files in PACT, you can choose **File Upload**. The **Attach File** field will appear. Click **Choose File** and navigate to the file you want to upload.
- **Module:** Specify the module to which this source object applies.
- **Function:** Specify the function this source object supports.
- **Image:** This function is TBD.
- **Item:** This is set to the current Item by default.
- **Link High-Level Requirements/Link Low-Level Requirements Buttons:** This is described in section 0 that follows.

After you fill in all the pertinent fields, you can Select **Save Source Code** to create the source code or **Back** to exit without saving the Source Code.

6.2 Editing a Source Code Record

To edit a Source Code record, from the **Source Code List Page**, click the  icon for a specific Source Code or choose the **Edit** button from the **Source Code View Page**. Editing a Source Code record is identical to creating a new Source Code record and all the fields are the same.



*Note that any time you click **Save Source Code** -- even if you do not change anything -- PACT increments the version. This allows for tracking a history of the source code*

record. Also note that this versioning is not the same as the strict versioning that PACT supports for documents that are within the PACT environment.

6.3 Deleting Source Code

From the **Source Code List Page**, Clicking the  or  icons for a specific Source Code record will delete that Source Code record. Before deleting a Source Code record, a confirmation dialog will appear asking if you are sure you want to delete the Source Code. You have to select **OK** to complete the deletion.

The difference between **Mark as Deleted** and **Delete** is that **Mark as Deleted** will leave the source code record but will “strikeout” the contents. This permits you to maintain a record that the Source Code existed at one time but was deleted. **Delete** will remove any trace of the Source Code. You should use the **Delete** capability with caution.

6.4 Linking Source Code to Requirements

When adding or editing a Source Code record, you have the option to link the Source Code to one or more High-Level or Low-Level Requirements. You will see these buttons at the bottom of the **New/Edit Source Code Page**. Both types of linking are similar.

To link to Low-Level Requirements, click the **Link Low-Level Requirements** button. This expands the page to include a linking section as follows.

Choose Hardware/Software Item in which to find Low-Level Requirements

CommSoftware
Ethernet_fpga
I/O_Software
MHW_TO

Search:

	Hardware/Software Item	Requirement ID	Description	Derived
<input type="checkbox"/>	I/O_Software	I/O_Software-LLR-001	All A429 Transmitters shall transmit at a baud rate of 1ghz.	<input type="checkbox"/>
<input type="checkbox"/>	I/O_Software	I/O_Software-LLR-002	The design shall do things it must do.	<input type="checkbox"/>

Save Low-Level Requirement Links

Save Source Code **Back** **Undo**

Figure 6-3. Link Source to Low-Level Requirements

First, select the item to which you want to link requirements. This is set to the current item by default, but if your source code supports requirements from another item also you can establish this link. The list of requirements changes if you change the item.

In the Requirements list area, click the checkbox before one or more Low-Level Requirements. Clicking the checkbox at the top of the list (in the heading row) will select or clear all checkboxes, depending on if it is checked or not.

You can select one or more requirements to which you want to link the Source Code. The click **Save Low-Level Requirement Links** to link the Source Code to the Low-Level requirement(s).

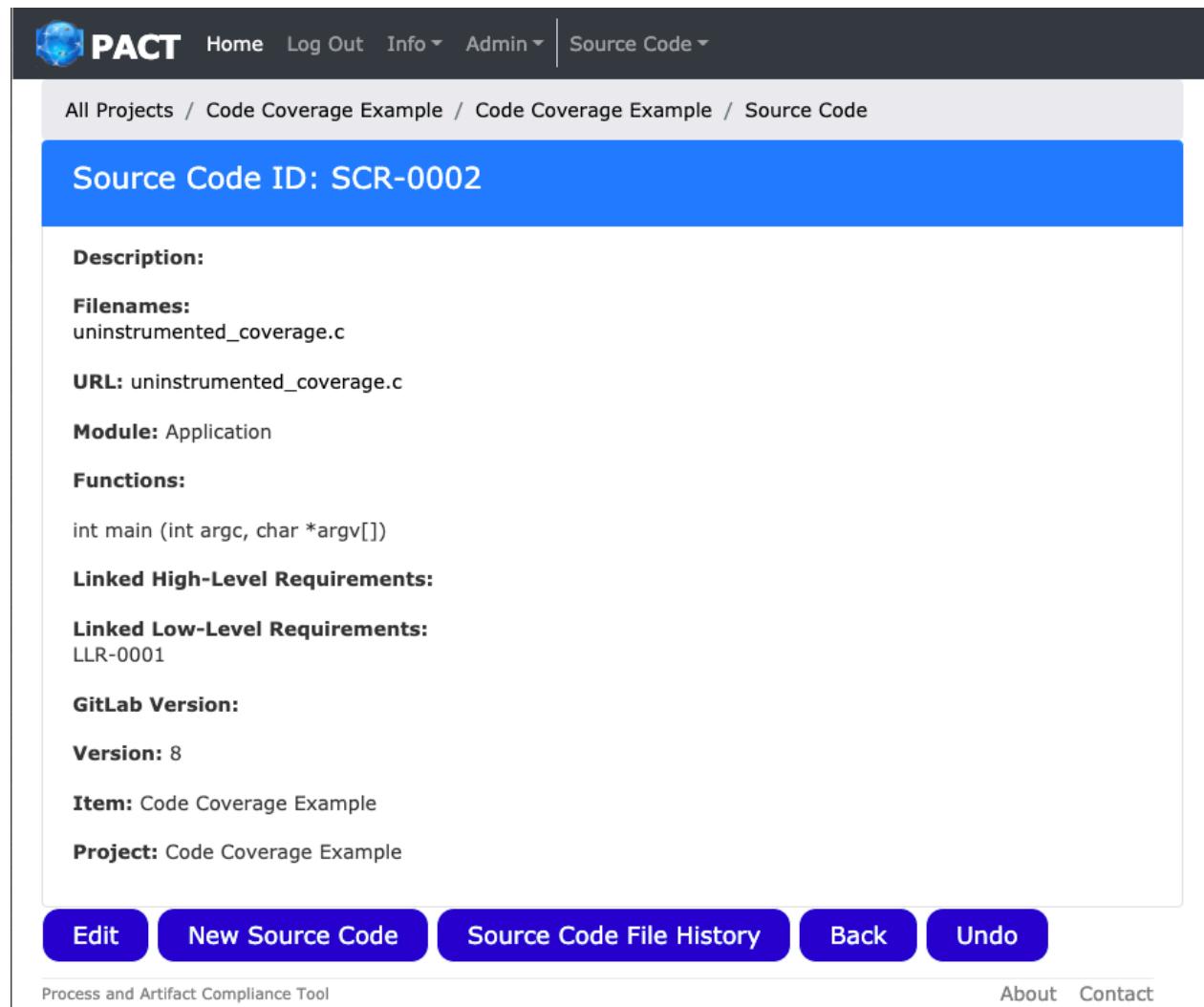
Make sure you this button before you click **Save Source Code** or your links will not be saved.

Follow a similar process (choosing the **Select High Level Requirements** button instead) to link Source Code to High-Level Requirements.

When you link Source Code to Requirements, this tells PACT how to generate the RTM (Requirements Traceability Matrix). See Section 9 for more information.

6.5 Viewing Source Code

To view a Source Code record from the **Source Code List**, click the  icon for a specific Source Code record or choose the  button from the **Source Code Edit Page**.



The screenshot shows the PACT application interface for viewing source code. At the top, there is a navigation bar with the PACT logo, Home, Log Out, Info, Admin, and Source Code dropdowns. Below the navigation bar, the URL path is shown as All Projects / Code Coverage Example / Code Coverage Example / Source Code. The main content area has a blue header bar with the text "Source Code ID: SCR-0002". The page displays various details about the source code, including:

- Description:** uninstrumented_coverage.c
- Filenames:** uninstrumented_coverage.c
- URL:** uninstrumented_coverage.c
- Module:** Application
- Functions:**

```
int main (int argc, char *argv[])
```
- Linked High-Level Requirements:**
- Linked Low-Level Requirements:** LLR-0001
- GitLab Version:**
- Version:** 8
- Item:** Code Coverage Example
- Project:** Code Coverage Example

At the bottom of the page, there are several buttons: Edit (highlighted), New Source Code, Source Code File History, Back, and Undo. The footer contains links for Process and Artifact Compliance Tool, About, and Contact.

Figure 6-4. View Source Code Page

6.6 Viewing Source Code History

If the source code files have been directly attached to the source code record in PACT, the history of the changes can be viewed. The files can also be diffed if they are text files. To view the history of a source code record, click the **Source Code File History** button while in the Source Code record show page. This will open the Source Code History page.

The screenshot shows the 'Source Code File History' page. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and a search bar labeled 'Search:'. Below the navigation bar, the URL path is shown as All Projects / Code Coverage Example / Code Coverage Example / Source Code. The main content area is titled 'Source Code File History'. A table lists four versions of the file 'uninstrumented_coverage.c' for source code record SCR-0002. The columns in the table are: Source Code ID, Description, Filenames, GitLab Version, Version, and two buttons for each row: 'Show' and 'Download'. The rows are as follows:

Source Code ID	Description	Filenames	GitLab Version	Version	Show	Download
SCR-0002		uninstrumented_coverage.c		2	Show	Download
SCR-0002		uninstrumented_coverage.c		4	Show	Download
SCR-0002		uninstrumented_coverage.c		6	Show	Download
SCR-0002		uninstrumented_coverage.c		8	Show	Download

Below the table, there are several checkboxes with labels: 'Side By Side', 'Show C Function', 'Ignore Case', and 'Ignore White Space'. At the bottom, there are two buttons: 'Diff Files' and 'Back'. The footer of the page includes links for 'About' and 'Contact', and a copyright notice: 'Process and Artifact Compliance Tool © ACS 2021. All rights reserved.'

Figure 6-5. Source Code History Page

To view a diff of two different versions of the file. Two (and only two) files must be selected in the Source code history page. When two files have been selected the **Diff Files** button will be selectable. Additional check boxes may be selected before clicking the **Diff Files** button to control the display of the diff. If the **Side by Side** checkbox is selected the diff will be side by side rather than an in-line diff. If the **Show C Function** checkbox is selected and the file is a C File, the diff will display the function name where each difference occurs. **Ignore Case** will ignore case when comparing the files and **Ignore White Space** will ignore white space (spaces, tabs, etc.) when comparing the files.

When **Diff Files** is clicked, the diff of the versions will be displayed.

The screenshot shows a web-based interface for comparing two source code files. At the top, there's a navigation bar with the PACT logo, Home, Log Out, Info, and Admin options. Below the navigation is a breadcrumb trail: All Projects / Code Coverage Example / Code Coverage Example / Source Code. The main title is "Diff Source Code Files". The content area displays a diff of two C source code files. The diff highlights changes with red and green colors. A "Close" button is at the bottom left, and links for About and Contact are at the bottom right.

```
22c22
<     printf("In function call.\n");
---
>     fprintf(stderr, "In function call.\n");
32c32
<     printf("Before exit.\n");
---
>     fprintf(stderr, "Before exit.\n");
110c110
<     printf("In if\n");
---
>     fprintf(stderr, "In if\n");
137c137
<         printf("In for: %d\n", i);
---
>         fprintf(stderr, "In for: %d\n", i);
158c158
<             printf("In switch: case 1\n");
---
>             fprintf(stderr, "In switch: case 1\n");
164c164
<                 printf("In switch: break\n");
---
>                 fprintf(stderr, "In switch: break\n");
168c168
<                     printf("In switch: case 2\n");
---
>                     fprintf(stderr, "In switch: case 2\n");
174c174
<                         printf("In switch: break\n");
---
>                         fprintf(stderr, "In switch: break\n");
180c180
<                             printf("In switch: default\n");
---
>                             fprintf(stderr, "In switch: default\n");
```

Figure 6-6. Source Code Diff Page

6.7 Exporting Source Code

Exporting Source Code permits you to save the source code to a file that can be 1) cut-and-pasted into the appropriate <Hardware/Software> *Design Document* as part of your certification documentation set stored within PACT, or 2) potentially imported into another application.

You can export Source Code as an HTML document, PDF document, Comma Separated Value (CSV) file or an Excel File (XLS).

To export Source Code, from the **Source Code List Page**, click the **Export** button. This opens the **Export Source Code Page** (in a new browser tab).

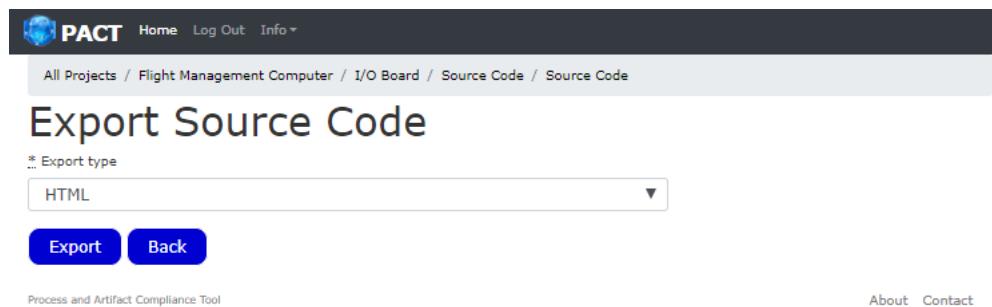


Figure 6-7. Export Source Code Page

Choose the type of export you would like (HTML, PDF, CSV or XML) and click the **Export** button. Depending on format, the information will either display in this new page or a file will download to your computer. Close the **Export Source Code Page** to ensure you do not end up with multiple PACT session tabs.

6.8 Importing Source Code

Importing Source Code permits you to load the Source Code for a Hardware/Software Item from another application. You can import Source Code from a Comma Separated Value (CSV) file or Excel File (XLS or XLSX).

The file should only contain data (**no extra headers or extraneous entries**) with lines in the format as follows (on one line):

```
id,codeid,full_id,file_name,module,function,derived,derived_justification,high_level_requirement_associations,low_level_requirement_associations,url_type,url_description,url_link,version,item_id,project_id,created_at,updated_at,organization,archive_id,description,soft_delete,file_path,content_type,file_type,revision,draft_version,revision_date,upload_date,external_version
```

To import Source Code, from the **Source Code List**, click the **Import** button.

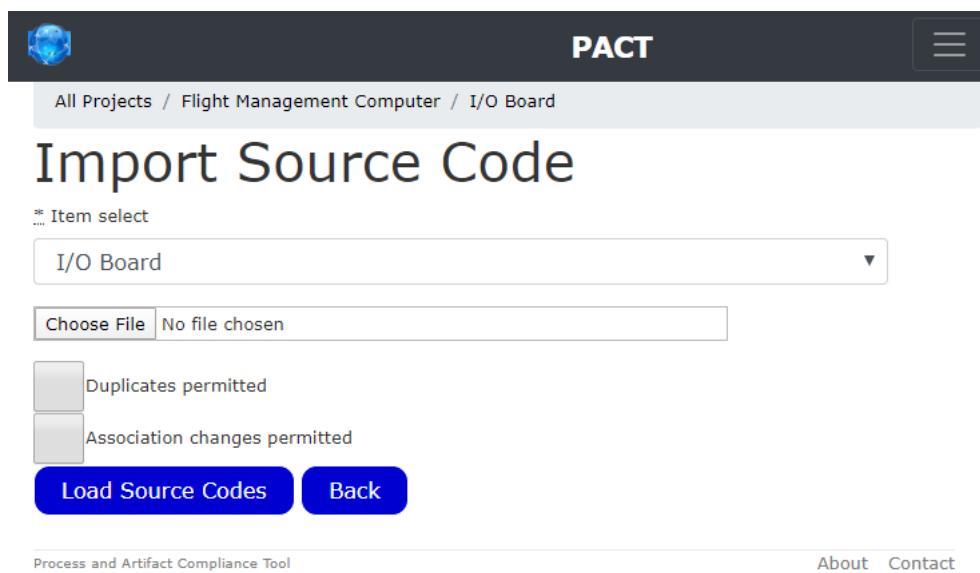


Figure 6-8. Import Source Code Page

To Import Source Code, perform the following tasks:

- 1 Choose the **File** you would like to import (in the format shown previously).
- 2 Click **Duplicates Permitted** if you want to permit duplicate source code IDs to be imported. If you do not check Duplicates Permitted and a Source Code record already exists with the same ID as in the file when the file is imported, it will cause an error and the file will not be imported.
- 3 Click **Association Changes Permitted** if you want to permit changes to Low-Level Requirement or High-Level Requirement associations. If you do not check this and a Source Code record already exists with the same ID as in the file when the file is imported and it changes the Low or High-Level Requirement associations, it will cause an error and the file will not be imported.

6.9 Zip Source Code Files

You can export all the source code files into a Zip archive. To save the files into a Zip archive click the **Zip Source Code** button. The following screen will appear (see next page):

The screenshot shows the 'Zip Source Code' page in the PACT interface. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Back. Below that is a breadcrumb trail: All Projects / Flight Management Computer / Test / Source Code. The main title is 'Zip Source Code'. A text input field shows 'Test-source_codes.zip'. Below it are two buttons: 'Create Source Code Package' (highlighted in blue) and 'Back'. A table lists files with checkboxes for selection:

	Code ID	Full ID	File Name	Module	Function
<input type="checkbox"/>	1	libhash/Makefile	libhash-Makefile		clean install uninstall
<input type="checkbox"/>	2	libhash/Makefile.nmake	libhash-Makefile.nmake		clean
<input type="checkbox"/>	3	libhash/README.md	libhash-README.md		
<input type="checkbox"/>	6	libhash/test_libhash.c	libhash-test_libhash.c		static double epoch_double(void) int main(int argc, char * argv[])
<input type="checkbox"/>	7	libhash/version.sh	libhash-version.sh		
<input type="checkbox"/>	4	libhash/libhash.c	libhash-libhash.c		static uint32_t hash_dobbs(const char *key, size_t len) static uint32_t hash_murmur(const char *key, size_t len) static uint32_t hash_city(const char *key, size_t len) static uint32_t hash_spooky(const char *key, size_t len) uint32_t hash_gen32(const char *key, size_t len) static struct hash_item *hash_lookup(struct hash *h, const char *key, uint32_t *hash) struct hash *hash_create(int bucket) void hash_destroy(struct hash *h) void hash_set_destroy(struct hash *h, void (*destory)(void *val)) void *hash_get(struct hash *h, const char *key) int hash_set(struct hash *h, const char *key, void *val) int hash_del(struct hash *h, const char *key) void *hash_get_and_del(struct hash *h, const char *key)
<input type="checkbox"/>	5	libhash/libhash.h	libhash-libhash.h		void hash_set_destroy(struct hash *h, void (*destory)(void *val));

Process and Artifact Compliance Tool (C)
Airworthiness Certification Services, LLC
PMB#224
5500 Olympic Drive
Suite H-105
Gig Harbor Wa 98335
Phone/fax: 425-427-1956

About Privacy and Security Contact

Figure 6-9. Zip Source Code Page

Enter the filename to save the Zip Archive into. Select the files you want to include in the Zip archive (check the boxes next to each file) then click the **Create Source Code Package** button. The files will be stored in the archive and the archive will be downloaded.

6.10 Scanning GitHub or GitLab Files

If your Source Code is version controlled in a GitHub or GitLab repository, you can automatically scan the Source Code Records into PACT. To do this, you first need to set up your GitHub or GitLab Credentials in PACT if you have not done so already. To do this choose **Info** from the menu at the top of any page and then choose **Edit GitHub Credentials** from the dropdown menu.



Note: The process for scanning files from either GitHub or GitLab is virtually identically. We are showing the process for GitHub here.

The process for scanning files from either GitHub or GitLab is virtually identically. The GitHub process is shown here.

The screenshot shows the 'Editing Github Credentials' page in the PACT software. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and GitHub Access. Below the navigation bar, the title 'Editing Github Credentials' is displayed. There are four input fields: 'Username' (empty), 'Password' (empty), 'Token' (empty and highlighted with a blue border), and 'User' (set to 'Paul Carrick'). Below the input fields are four buttons: 'Save Credentials' (highlighted with a blue background), 'Show', 'Back', and 'Undo'. At the bottom of the page, there are links for 'Process and Artifact Compliance Tool', 'About', and 'Contact'.

Figure 6-10_1. Edit GitHub Credentials Page

Enter either your GitHub Username and Password or the token you setup for access in GitHub and click **Save Credentials**.

Once your credentials are setup in PACT, you can click **Scan GitHub** in the **Source Code List**.

Scan for Source Files

Repository

Demo Project

Branch

master

Folder

libhash

Scan Back

Process and Artifact Compliance Tool

About Contact

Figure 6-10_2. Scan GitHub Screen

Select the Repository, Branch and Folder you want to scan press **Scan**. This brings you to the **Select Files Page**.

<input type="checkbox"/>	Filename
<input type="checkbox"/>	libhash/Android.mk
<input type="checkbox"/>	libhash/Makefile
<input type="checkbox"/>	libhash/Makefile.nmake
<input type="checkbox"/>	libhash/README.md
<input checked="" type="checkbox"/>	libhash/libhash.c
<input checked="" type="checkbox"/>	libhash/libhash.h
<input checked="" type="checkbox"/>	libhash/test_libhash.c
<input checked="" type="checkbox"/>	libhash/version.sh

attach Files

Done Back

Process and Artifact Compliance Tool About Contact

Figure 6-10-3. Select Files Page

You may click the checkbox before one or more files. Clicking the checkbox at the top of the list will select or clear all checkboxes. Clicking **Done** will import the files into the **Source Code List Page**. **Back** will return without making any changes.

For GitLab you may automatically attach the file to the source code record. To attach the file automatically click the Attach Files checkbox before clicking **Done**. If the files are C/C++ files the function names will be scanned and added to the source code record.

Source Code ID	Description	File Name	URL	Module	Functions	Low-Level Requirements	High-Level Requirements	Version				
APC_PIC_SW-APC_A429_FPGA-SC-1		libhash/libhash.c	https://gitlab.firebaseioconsultants.com/palmos/demo-project/blob/master/libhash/libhash.c		<pre>static uint32_t hash_dob(const char *key, size_t len) static uint32_t hash_murmur(const char *key, size_t len) static uint32_t hash_onyx(const char *key, size_t len) static uint32_t hash_poor(const char *key, size_t len) uint32_t hash(const char *key, size_t len)</pre>							
APC_PIC_SW-APC_A429_FPGA-SC-2		libhash/libhash.h	https://gitlab.firebaseioconsultants.com/palmos/demo-project/blob/master/libhash/libhash.h		void hash_set_destroy(struct hash *h, void (*destroy)(void *val));							
APC_PIC_SW-APC_A429_FPGA-SC-3		libhash/test.libhash.c	https://gitlab.firebaseioconsultants.com/palmos/demo-project/blob/master/libhash/test.libhash.c		static double epoch_double(void) int main(int argc, char * argv[])							
APC_PIC_SW-APC_A429_FPGA-SC-4		libhash/version.sh	https://gitlab.firebaseioconsultants.com/palmos/demo-project/blob/master/libhash/version.sh									

New Export Import Scan GitHub Scan GitLab Renumber Set Up Coverage Process Profile New Baseline View Baselines Back Undo

Process and Artifact Compliance Tool About Contact

Figure 6-10_4. Source Code List after Importing Files from GitLab.

6.11 Renumbering Source Code

Renumbering Source Code permits renumbering the Source Code starting at a given value and incrementing a specific value.

All Projects / Requirements Tracing / Software Item / Source Code

* Start ...

* Increment ...

* Leading zeros ...

3

Renumber Back

Process and Artifact Compliance Tool (C)
Airworthiness Certification Services, LLC
PMB#224
5500 Olympic Drive
Suite H-105
Gig Harbor Wa 98335
Phone/fax: 425-427-1956

About Privacy and Security Contact

Figure 6-11. Renumber Source Codes.

To renumber Module Descriptionsl Requirements, enter a starting number and an increments then and then click **Renumber** from the **Source Code Renumbner** Page.



Renumbering Source Code is something that should be done only under extreme circumstances and/or prior to the first baseline/formal review of the requirements. Otherwise, requirements history and traceability can be adversely affected.

6.12 Baselining Source Code Records

Baselining Source Code saves a read-only copy of the current Source Code Records along with any associated requirements. To create a new baseline, perform the steps below.

- 1 Click the **New Baseline** button. This brings up the **New Source Code Baseline Page**, like in the example figure that follows.

The screenshot shows the 'New Source Code Baseline' page. At the top, there's a navigation bar with the PACT logo, Home, Log Out, Info, Admin, and Requirements Baseline. Below the navigation is a header 'All Projects'. The main section has a title 'New Source Code Baseline'. It contains several input fields: 'Name' (Project: Process and Artifact Compliance Tool baseline of Source Code), 'Baseline ID' (Source Code Baseline-0.1), 'Description' (Project: Process and Artifact Compliance Tool baseline of Source Code), 'Pact version' (1.7), 'Revision' (empty), 'Draft Version' (0.1), and 'Baselined At' (07/17/2020, 12:59:06 PM). At the bottom are three buttons: 'Create Requirements Baseline' (highlighted in blue), 'Back', and 'Undo'.

Figure 6-12. New Source Code Baseline Page

2 Enter the information on this page as follows:

- **Name:** This is the long name used to identify the Baseline. It is prefilled but may be changed.
- **Baseline ID:** This is the short identifier used to identify the Baseline. It is prefilled but may be changed.
- **Description:** This a description for the Baseline. It is prefilled but may be changed.
- **PACT Version:** This is version of PACT that was used to create the Baseline. It may not be changed.
- **Revision:** This a revision that is associated with the Baseline. It does not need to be filled in and may be left blank.
- **Draft Version:** This the version for the Baseline. It is prefilled but may be changed. It defaults to the next available floating-point number but may be an integer or a floating-point number. A string can be entered but then the auto-incrementing will be disabled.
- **Baselined At:** This is the date and time of the baseline and may not be changed.

3 You can Select [Create Requirements Baseline](#) to create the Source Code Baseline or [Back](#) to exit without saving the Source Code Baseline. Creating the baseline may take a few minutes based on how many codes there are.

6.13 Viewing Source Code Baselines

Clicking the [View Baselines](#) button brings up the **Source Code Baseline List Page**, like in the example figure that follows.

Name	Identifier	Description	Revision	Draft Version	Baselined At							
Project: Process and Artifact Compliance Tool baseline of Source Code.	Source Code Baseline-0.1	Project: Process and Artifact Compliance Tool baseline of Source Code.		0.1	2020-07-17 01:14PM PDT	View Contents	Show	Edit	Delete			

Figure 6-13_1. New Source Code Baseline Page

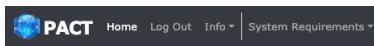
On the **Source Code Baselines List Page**, you can do the following:

- View the contents of the Baseline
- View the information about the Baseline.

- Edit the information about existing Baseline
- Delete a Baseline
- Create a New Source Code Baseline

To view the contents of a Source Code Baseline, from the **Source Code Baselines List Page**,

click the  icon for a specific Baseline. This will take you into the contents of the Baseline. In the Baseline you will see a “snapshot” of the codes at a given time. The codes in this view cannot be changed. You will know that you are viewing a baseline by the word **Baseline** in the



Breadcrumbs bar    . You can navigate in the Baseline in exactly the same manner in exactly the same way as with all other projects. The only differences are that you are seeing the codes for a given time period and that you cannot change anything. To leave the Baseline return to the PACT home screen by Clicking Home from the Menu or clicking All Projects in the *Breadcrumbs* bar.

To view a Source Code Baseline record, from the **Source Code Baselines List Page**, click the  icon for a specific Baseline. This provides a Page showing all the information related to the Baseline.

Details:

Full: System Requirements Baseline-0.1
Description: Project: Flight Management Computer baseline of System Requirements.
Revision:
Version: 0.1
Baselined at: 2020-07-15 11:16AM PDT
Pact version: 1.7
Baseline type: System Requirements
Organization: Patmos Engineering Services
Project: Flight Management Computer
Item:

Edit New Requirements Baseline Back

Figure 6-13_2. Viewing a Specific Source Codes Baseline

Clicking the  icon for a given Baseline will permit you to edit the information for that Baseline. This Page works exactly like the New Source Code Baseline Page. Note that you can only change specific information about a Baseline and that the contents of the baseline can never be changed.

From the **Source Code Baselines Page**, clicking the  icon for a specific Source Code Baseline will delete that Source Code baseline. Before deleting a Source Code Baseline, a

confirmation dialog will appear asking if you are sure you want to delete the Source Code. You must “OK” the dialog to permit the operation.

To create an additional Baseline, you can click [New Requirements Baseline](#). The process is identical to the process in Section 6.12.

6.14 Utilizing Source Code Coverage



Note: This section only applies to those organizations who have purchased the PACT Code Coverage option and currently only applies to Software Items (i.e., not Hardware). THIS FUNCTION IS CURRENTLY IN BETA TESTING, NOT FULLY RELEASED.

Assessing verification/test coverage is an important part of safety-critical programs. In these programs you develop and run requirements-based tests on the code you develop. This ensures the requirements are implemented in the code and tested. You may also be required to run coverage metrics on the code during this process. This ensures that requirements-based testing is activating all the code. If code is not activated during testing, this means it is extraneous for some reason (e.g., dead code) and needs to be removed or justified to ensure it serves a purpose and does not impact safety.

If you have purchased the Code Coverage option for PACT, you will see additional buttons in the Source Code List.

These buttons are: [Setup Coverage](#) [Process](#) [Profile](#) which are necessary to perform the several steps involved in running code coverage.

6.14.1 Understanding Code Coverage in PACT

Code coverage in PACT entails setting Code Marks in your code that records which lines of the code are executed and optionally when they were hit. It can be used to identify coverage of the test cases (to see how thoroughly they test the code), dead or extraneous code that should be removed, and potentially missing requirements or test cases.

Optionally code coverage can trace the path of execution through the code. This entails instrumenting your code with Macros that record when/if a particular line is executed, adding the code to PACT, executing the code in your runtime environment, and then uploading the results into PACT which will then display a profile of your code.

6.14.2 Instrumenting Code

While PACT can instrument any source code that is based on text files and can support any computer language that permits a statement like CMARK(n) or cmark(n), this section describes how to instrument a C file as an example. From this, you should be able to extrapolate what to do to instrument your own code.

In order to perform code coverage in PACT, the C code must implement **Code Checkmarks**. You will need to edit your source code files to add **Code Checkmarks**. A **Code Checkmark** records if/when specific lines in the source code are executed. This is normally done via a C macro **CMARK...** The **CMARK** macro will record when/if a **CMARK...** macro is executed. In most cases this is done via writing a string to **STDOUT** that then can be redirected into a file, which is then imported into PACT. PACT supports several different formats for the lines in a code coverage file so you can see the coverage percentage and the lines that were executed. Each option, which has pros and cons, is presenting in the following subsections.

6.14.2.1 CMARK Option

The simplest format is:

number

number

number

For Example:

1

2

3

This records that a particular **CMARK** was executed.

You can supply a file (**cmark.h**) that can be included in your source code files. The header file requires no other included files. It defines a macro **#define CMARK(n)** .

Add the line **CMARK(n)** at each point you want to record code coverage.

For example:

```
#include "cmark.h"

int main (int argc, char *argv[])
{
    int i;
    i = 1; CMARK(0);

    if (argc > 1)
    {
        i +=1; CMARK(1);
```

```
...  
}
```

The advantage is that this is the simplest implementation and requires the least impact on your code. The disadvantage is that it does not record when the Code Mark was hit, so it can only be used for coverage but not tracing.

6.14.2.2 CMARK_NANOSECONDS Option

Another format is as follows:

number,seconds.nanoseconds
number,seconds.nanoseconds
number,seconds.nanoseconds

For Example:

0,1577744431.470873000
1,1577744431.471032000
2,1577744431.471043000

This records that a particular CMARK was executed at a given time in seconds and *.nnnn* milliseconds from the Unix/Linux Epoch.

We can supply a file (**cmark_nanoseconds.h**) that can be included in your source code files. The header file requires **time.h** which in turn defines **struct timespec** and implements **clock_gettime(CLOCK_REALTIME, struct timespec*)**.

It defines a macro: **CMARK_NANSECONDS (#define CMARK_NANSECONDS (n))**.

You simply add the line **CMARK_NANSECONDS (n)** at each point you want to record code coverage.

For example:

```
#include "cmark_nanoseconds.h"

int main (int argc, char *argv[])
{
    int i;
    i = 1; CMARK_NANOSECONDS (0);

    if (argc > 1)
    {
        i +=1; CMARK_NANOSECONDS (1);
```

```
...  
}
```

The advantage is that this is the most precise definition and can be used for coverage tracing and profiling. The disadvantage is that `clock_gettime(CLOCK_REALTIME, struct timespec*)` must be available in your runtime environment.

6.14.2.3 CMARK_SECONDS Option

Another format is:

number,seconds

number,seconds

number,seconds

For example:

0,1577745741

1,1577745741

2,1577745741

3,1577745741

This records that a particular CMARK was executed at a given time in seconds from the system's Epoch.

We can supply a file (`cmark_seconds.h`) that can be included in your source code files. The header file requires `time.h` which in turn defines `time_t` and implements `time(time_t *)`. It defines a macro: `CMARK (#define CMARK(n))`.

You add the line `CMARK_SECONDS(n)` at each point you want to record code coverage.

For example:

```
#include ""cmark_seconds""  
  
int main (int argc, char *argv[])  
{  
    int i;  
    i = 1; CMARK_SECONDS(0);  
  
    if (argc > 1)  
    {  
        i +=1; CMARK_SECONDS(1);  
        ...  
    }  
}
```

```
}
```

The advantage is that this does not require `clock_gettime` but it does require `time()`. The disadvantage is that the resolution is only in seconds.

6.14.2.4 CMARK_STRFTIME Option

Another format is:

`number,YYYY-mm-ddTHH:MM:SS`

`number,YYYY-mm-ddTHH:MM:SS`

`number,YYYY-mm-ddTHH:MM:SS`

For example:

`0,2019-12-30T14:21:02`

`1,2019-12-30T14:21:02`

`2,2019-12-30T14:21:02`

This records that a particular CMARK was executed at a given time.

You can supply a file (`cmark_strftime.h`) that can be included in your source code files. The header file requires `time.h` and that `time.h` defines `time_t` and implements `time(time_t *)`. It also requires `stdio.h` which defines a macro: `CMARK (#define CMARK(n))`.

Add the line `CMARK(n)` at each point you want to record code coverage.

For example:

```
#include "cmark_strftime.h"

int main (int argc, char *argv[])
{
    int i;
    i = 1; CMARK_STRFTIME (0);

    if (argc > 1)
    {
        i +=1; CMARK_STRFTIME (1);
        ...
    }
}
```

The advantage is that the output is human readable, but it does require `time()` and `stdio.h`.
The disadvantage is the resolution is only in seconds.

6.14.2.5 CMARK_BITMAP Option

A final way to record hits is to use a bitmap:
number

For Example:

0x800000003fe3deff

This records that a particular CMARK was hit by setting a bit for each CMARK hit.

Define:

```
#define CMARK_BITMAP(n, map) map |= 1LL << n;  
  
unsigned long long int bitmap;  
  
int main (int argc, char *argv[])  
{  
    int i;  
    i = 1; CMARK_BITMAP (0, bitmap);  
  
    if (argc > 1)  
    {  
        i +=1; CMARK_BITMAP (1, bitmap);  
        ...  
    }  
}
```

The advantage is that this does not require any external code or IO but you must provide a way to transfer the value to a file. The disadvantage is this method does not record the or how many times the mark is hit. Also depending on the system, you are limited to 64 CMARKS. However, you can use this in the case of an embedded system that has no IO.

6.14.3 Setting Up Coverage

First you will need to get the (un-instrumented) Source Code into PACT (see earlier sections that discussed how to do this).

Once you have the files in PACT, click on the **Setup Coverage** button from the **Source Code List**. This will record the CMARK statements in your source files before you compile and run the code on your target platform.

	Code ID	Full ID	File Name	Module	Function
<input type="checkbox"/>	SCR-0002	uninstrumented_coverage.c	uninstrumented_coverage.c	Application	int main (int argc, char *argv[])

Auto Instrument: **No** Code Mark Macro: **CMARK**

Done **Back**

Figure 6-14_1. Set Up Source Code Coverage Page

Choose the files you want to analyze coverage on by clicking the checkboxes on the far left.

If you want to have PACT automatically add the **CMARKs** to your source code, chose **Yes** for Auto Instrument otherwise leave it as **No**. If you chose to Auto Instrument your code you need to choose the CMARK type (see Section Instrumenting Code6.14.2). and click the **Done** button.

If you have chosen to auto instrument your code, you will need to download the auto instrumented files. After you Setup Coverage you will find the instrument files in a folder called **Instrumented Code** in under the **Documents List**. You can download the files by clicking on the **Document ID** in the **Document List**.

Next, compile and run the files on the target system. When you run the files redirect the STDOUT to a log file (or files). This records the lines and times that the specific code marks were executed.

6.14.4 Processing Log Files

In order to evaluate the coverage of your source code you will need to process the log files you created in the previous step. To process the log files, click on the **Process** button from the **Source Code List**. This will process the log file and record the results.

The screenshot shows the PACT interface for processing code run results. At the top, there's a navigation bar with the PACT logo, Home, Log Out, Info, and Admin options. Below it, a breadcrumb trail shows All Projects / Code Coverage Example / Code Coverage Example / Source Codes. The main title is "Process Code Run Results" under "Source Codes". A table lists a single source code entry:

<input type="checkbox"/>	Code ID	Full ID	File Name	Module	Function
<input type="checkbox"/>	SCR-0002	uninstrumented_coverage.c	uninstrumented_coverage.c	Application	int main (int argc, char *argv[])

Below the table is a file input field labeled "Choose File" with "No file chosen". At the bottom are two buttons: "Process Code Run Results" (highlighted in blue) and "Back".

Figure 6-14_2. Process Code Run Results Page

Choose the files that were run to create the log file by clicking the checkboxes before them and then chose the log file to upload. Finally click the **Process Code Run Results** button. A new **Run Results Profile Page** will appear.

The screenshot shows the PACT interface for the Run Results Profile page. At the top, there's a navigation bar with the PACT logo, Home, Log Out, Info, and Admin options. Below it, a breadcrumb trail shows All Projects / Process and Artifact Compliance Tool / Code Coverage Example / Code Coverage Example / Source Codes. A green success message says "Run results successfully processed." The main title is "Run Results Profile". A table displays coverage details:

Source Code ID	File Name	URL	Code Marks	Unique Code Mark Hits	Total Hits	Coverage
SCR-0001	coverage_nano_seconds.c	coverage_nanoseconds.c	36	26	44	72.22%

At the bottom are two buttons: "Back" (highlighted in blue) and "About Contact".

Figure 6-14_3. Run Results Profile Page

In the **Run Results Profile Page**, you will then see the coverage of all the Code Checkmarks that were hit.

6.14.5 Evaluating Results

You can click on the number below Code Marks, Total Hits or Code Marks Not hit to see the Code Checkmarks.

The screenshot shows the PACT interface with the title "Code Check Marks". Below the title is a search bar labeled "Search:". A table lists 24 code checkmarks, all associated with Source code ID SCR-0002 and filename uninstrumented_coverage.c. The table columns are: Checkmark ID, Source code ID, Filename, Line number, and Statement. The statements listed are C code snippets related to function calls, exits, loops, and variable assignments.

Checkmark ID	Source code ID	Filename	Line number	Statement
197	SCR-0002	uninstrumented_coverage.c	22	printf("In function call.\n");
198	SCR-0002	uninstrumented_coverage.c	27	return;
199	SCR-0002	uninstrumented_coverage.c	32	printf("Before exit.\n");
200	SCR-0002	uninstrumented_coverage.c	33	exit(0);
201	SCR-0002	uninstrumented_coverage.c	40	for (int i = 1; i < argc; i++)
202	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
203	SCR-0002	uninstrumented_coverage.c	48	test_if = TRUE;
204	SCR-0002	uninstrumented_coverage.c	49	test_while = TRUE;
205	SCR-0002	uninstrumented_coverage.c	50	test_for = TRUE;
206	SCR-0002	uninstrumented_coverage.c	51	test_do_while = TRUE;
207	SCR-0002	uninstrumented_coverage.c	52	test_switch = TRUE;
208	SCR-0002	uninstrumented_coverage.c	53	test_question_mark = TRUE;
209	SCR-0002	uninstrumented_coverage.c	54	test_break = TRUE;
210	SCR-0002	uninstrumented_coverage.c	55	test_default = TRUE;
211	SCR-0002	uninstrumented_coverage.c	56	test_return = TRUE;
212	SCR-0002	uninstrumented_coverage.c	57	test_function_call = TRUE;
213	SCR-0002	uninstrumented_coverage.c	58	test_exit = TRUE;
214	SCR-0002	uninstrumented_coverage.c	62	test_if = TRUE;

[Back](#)

Process and Artifact Compliance Tool

[About](#) [Contact](#)

Figure 6-14_3. Code Check Marks Page

You will then see the **Code Check Marks Page**. To see the hits, click on the Checkmark ID.

You can also click on the Source Code ID to see the Source Code Record or the Filename to see the file. To see the Code Checkmark Hit for a given Code Checkmark click on the number under Checkmark ID.

The screenshot shows a web application interface for PACT. At the top, there is a navigation bar with links for Home, Log Out, Info, and Admin. Below the navigation bar, the title "Code Checkmarks Hits" is displayed. To the right of the title is a search input field labeled "Search:". A table below the title lists the hits for checkmark ID 202. The table has columns: Checkmark ID, Hit At, Source code ID, Filename, Line number, and Statement. There are 14 rows in the table, each corresponding to a hit at a specific date and time, with the same source code ID (SCR-0002), filename (uninstrumented_coverage.c), line number (44), and statement (char *argument = &argv[i][1];).

Checkmark ID	Hit At	Source code ID	Filename	Line number	Statement
202	2020-01-24 09:43:55.344588000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344616000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344629000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344642000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344654000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344666000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344678000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344690000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344702000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344714000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344725000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344737000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];
202	2020-01-24 09:43:55.344813000	SCR-0002	uninstrumented_coverage.c	44	char *argument = &argv[i][1];

[Back](#)

Process and Artifact Compliance Tool

[About](#) [Contact](#)*Figure 6-14_4. Code Checkmarks Hits Page*

You will see a list of the times that the checkmark was Hit. For each hit you will see the Checkmark ID of the checkmark, the Source Code ID of the Source Code Record for that Checkmark, the Filename of the file, the Line Number that the checkmark is at and the Statement on that line.

7 Managing Test Cases & Test Procedures

Verification is a very important aspect of safety-critical development programs. It is a supporting process that runs parallel to design development. A big part of verification is the creation and running of requirements-based tests and linking those tests and their results back to their corresponding requirements. This section explains these terms and discusses how PACT can assist with managing test cases and procedures.

7.1 Understanding Test Cases and Test Procedures

It is important to understand Test Cases and Test Procedures and the role they play in the overall activity of design testing.

Each requirement for the design must be tested. This is done via the criteria specified by one or more *Test Cases* and executed by a corresponding *Test Procedure*.

A *Test Case* involves a set of test inputs, execution conditions, and expected results developed to verify compliance with a specific requirement.

A *Test Procedure* involves instructions for the set-up, execution and evaluation of one or more Test Cases.

For traceability purposes, a Test Case is linked to one or more requirements and a Test Procedure links to one or more associated Test Cases that it executes. These connections are established during the verification process and become part of the Requirements Traceability Matrix.

7.2 PACT Support of Test Cases & Procedures

Your scripts and other files associated with testing will exist in your external verification environment created and managed by the verification team. These files will be controlled in whatever manner your verification plans define. These test case files do not actually exist within the PACT environment (although Test Procedures can reference executable files).

What you are doing in PACT is describing the test cases and test procedures and linking them to each other and their associated requirements for traceability purposes. The descriptions can then be exported into a format for copy and paste into the appropriate document that includes the test case and test procedure information, such as a typical *Software Verification Cases, Procedures and Results* document required for a DO-178C program. The document capturing the test case and test procedure information itself will be reviewed and controlled appropriately within the PACT environment.

Test cases and test procedures, since they are informational only in the PACT environment, are version controlled in a simple way. Every time there's a change to a test case or procedure, the version increments by 1. You can also report on the history of these changes.

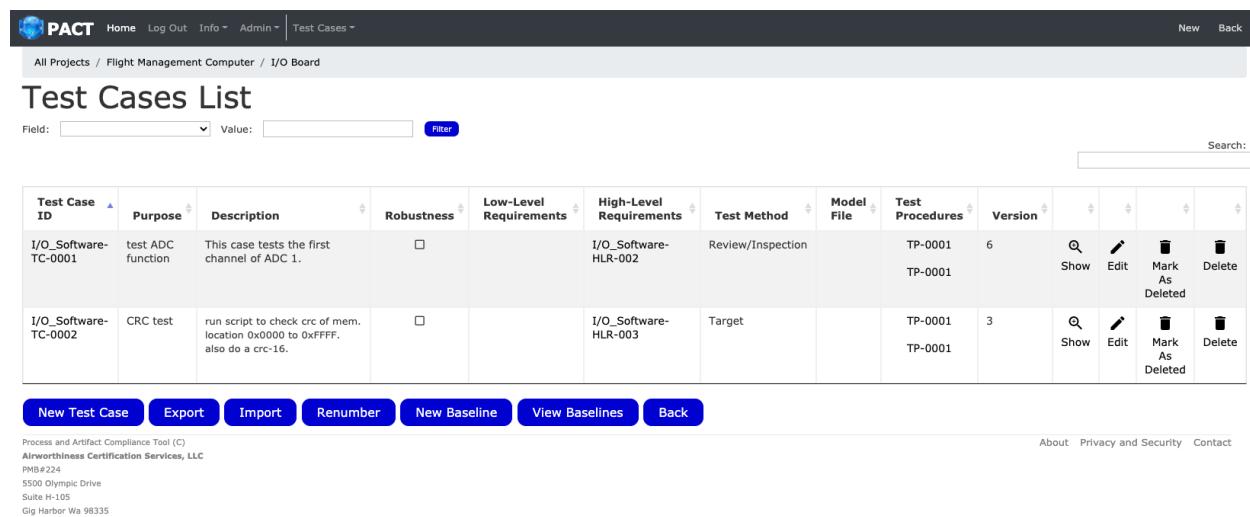
In the PACT environment, Test Cases and Test Procedures are associated with a Hardware/Software Item. You access the Test Case functionality by navigating to **Test Cases** under the appropriate Hardware/Software Item. You access the Test Procedures functionality by navigating to **Test Procedures** under the appropriate Hardware/Software Item. The next subsections discuss this further.

7.3 Managing Test Cases in PACT

Test Cases were described in Section 7.1. The following subsections describe how to manage Test Cases in PACT.

7.3.1 Listing Test Cases

Clicking the **Test Cases** link in the **Navigation** column of a specific **Hardware/Software Item** will display the **Test Cases List Page**.



The screenshot shows the 'Test Cases List' page for the 'I/O Board' hardware/software item. The page has a header with navigation links like Home, Log Out, Info, Admin, and Test Cases. Below the header, the URL is 'All Projects / Flight Management Computer / I/O Board'. The main content area is titled 'Test Cases List' and contains a table with two rows of test cases. The columns include: Test Case ID, Purpose, Description, Robustness, Low-Level Requirements, High-Level Requirements, Test Method, Model File, Test Procedures, Version, and several action buttons (Show, Edit, Mark As Deleted, Delete). The first test case is for 'I/O_Software-TC-0001' with purpose 'test ADC function' and description 'This case tests the first channel of ADC 1.'. The second test case is for 'I/O_Software-TC-0002' with purpose 'CRC test' and description 'run script to check crc of mem. location 0x0000 to 0xFFFF. also do a crc-16.'. At the bottom of the table are buttons for New Test Case, Export, Import, Rerumber, New Baseline, View Baselines, and Back.

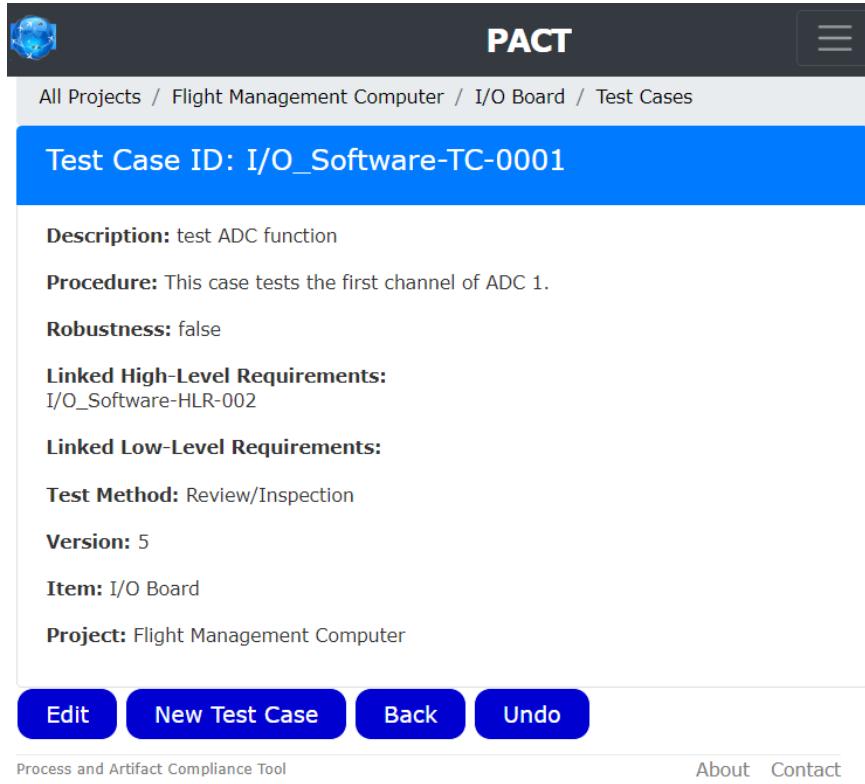
Test Case ID	Purpose	Description	Robustness	Low-Level Requirements	High-Level Requirements	Test Method	Model File	Test Procedures	Version	Show	Edit	Mark As Deleted	Delete
I/O_Software-TC-0001	test ADC function	This case tests the first channel of ADC 1.	<input type="checkbox"/>		I/O_Software-HLR-002	Review/Inspection		TP-0001 TP-0001	6				
I/O_Software-TC-0002	CRC test	run script to check crc of mem. location 0x0000 to 0xFFFF. also do a crc-16.	<input type="checkbox"/>		I/O_Software-HLR-003	Target		TP-0001 TP-0001	3				

Figure 7-1. Test Cases List Page

From here you can view Test Cases, add a new Test Case, show or edit an existing Test Case, mark a Test Case as deleted, delete a Test Case, link Test Cases to Requirements, export Test Case, import Test Cases, or renumber Test Cases. These activities are described in the following subsections.

7.3.2 Viewing a Test Case

To view a Test Case record, from the **Test Cases List Page**, click the  icon for a specific Test Case. This provides a Page showing all the information related to the Test Case.



The screenshot shows the PACT software interface. At the top, there is a navigation bar with the PACT logo and a search icon. Below the navigation bar, the breadcrumb navigation shows: All Projects / Flight Management Computer / I/O Board / Test Cases. The main content area has a blue header bar with the text "Test Case ID: I/O_Software-TC-0001". Below this, there are several data fields with their respective values:

- Description:** test ADC function
- Procedure:** This case tests the first channel of ADC 1.
- Robustness:** false
- Linked High-Level Requirements:**
I/O_Software-HLR-002
- Linked Low-Level Requirements:**
- Test Method:** Review/Inspection
- Version:** 5
- Item:** I/O Board
- Project:** Flight Management Computer

At the bottom of the content area, there are four buttons: Edit, New Test Case, Back, and Undo. Below the content area, there is a footer with links to Process and Artifact Compliance Tool, About, and Contact.

Figure 7-2. Viewing a Specific Test Case

7.3.3 Creating a New Test Case

Clicking the  button from the **Test Cases List Page** (or when viewing a specific test case) permits you to create a new Test Case.

New Test Case

* Test Case Number
3

* Test Case ID
I/O_Software-TC-0003

* Purpose

Description

Upload Image: Choose File No file chosen

Remove Image?
 Robustness?

Test Method: Review/Inspection

* Item: I/O Board

[Link High-Level Requirements](#) [Link Low-Level Requirements](#)

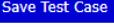
Save Test Case **Back** **Undo**

Figure 7-3. New Test Case Page

Fill in the fields of this page as follows:

- **Test Case Number:** This is assigned by the tool and is the next available number.
- **Test Case ID:** This required field is filled out by the tool to assign the test case a unique ID. The default ID is a combination of the Hardware or Software Item ID plus “TC” plus the next available number. For example, I/O_Software-TC-0003. While you can edit this ID, it is advisable that you do not unless your processes warrant a different naming/numbering scheme. In any case, ensure this field is always unique or PACT will issue an error.
- **Purpose:** Enter the purpose for the test case.
- **Description:** Enter a description for this Test Case. This should include set of inputs, conditions, expected results to achieve the required coverage criteria, and the pass/fail criteria.
- **Upload Image:** If there is an image file that helps describe the test you can upload it.
- **Robustness:** If the Test Case is related to Robustness click the Robustness Checkbox.

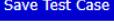
- **Test Method:** You may select one or more verification methods associated with the Test Case.
- **Item:** By default, this is set to the current Item.
- **Link High-Level Requirements/Link Low-Level Requirements** buttons: See section 7.3.6.

You can Select  to create the Test Case. When you save a new Test Case, PACT calls this Version 1. Click  if you want to exit without saving the Test Case.

7.3.4 Editing a Test Case

To edit a Test Case, from the **Test Cases List Page**, click the  icon for a specific Test Case or choose the  button from the **Test Case View Page**. Editing a Test Case is identical to creating a new Test Case and all the fields are the same. When you click the  button, the Test Case Version will increment.



Note that any time you click  -- even if you do not change anything – PACT increments the version. This allows for tracking a history of the test case record.

7.3.5 Deleting Test Cases

From the **Test Cases List Page**, Clicking the  or  icons for a specific Test Case will delete that Test Case. Before deleting a Test Case, a confirmation dialog will appear asking if you are sure you want to delete the Test Case. Click **OK** to confirm you want to perform the Delete action or click **Cancel** if you do not want to continue.

The difference between **Mark as Deleted** and **Delete** is that **Mark as Deleted** will leave the test case but empty its contents replacing it with “deleted.” This permits you to maintain a record that the Test Case existed at one time but was deleted. **Delete** will remove any trace of the Test Case.

7.3.6 Linking Test Cases to Requirements

When you add or edit a Test Case, you have the option to link the Test Case to one or more High-Level or Low-Level Requirements. You will see these buttons at the bottom of the **New/Edit Test Case Page**.

To link to High-Level Requirements, click the  button.

To link to Low-Level Requirements, click the  button.

Both types of linking are similar. The example that follows shows linking Test Cases to High-Level Requirements. If you click the **Link High-Level Requirements** button, the page will expand to include a new linking section as follows.

The screenshot shows a user interface for linking test cases to high-level requirements. At the top, there's a dropdown menu labeled "Choose Hardware/Software Item in which to find High-Level Requirements" with options like AHM, CommBrd_fpga, and CommSoftware. Below this is a search bar labeled "Search:". A large table lists requirements, each with a checkbox column for linking. The columns are: Hardware/Software Item, Requirement ID, Description, Safety, Robustness, and Derived. The table contains the following data:

Hardware/Software Item	Requirement ID	Description	Safety	Robustness	Derived
I/O_Software	I/O_Software-HLR-001	Test	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I/O_Software	I/O_Software-HLR-002	the i/o board shall tx and rx 1553 data.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I/O_Software	I/O_Software-HLR-003	the i/o board shall do more stuff.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I/O_Software	I/O_Software-HLR-004	the board shall never use req ID in this fashion.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
I/O_Software	I/O_Software-HLR-005	The I/O software shall have 4 A429 Tx channels.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I/O_Software	I/O_Software-HLR-006	All A429 channels shall operate at a 1ghz baud rate.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I/O_Software	I/O_Software-HLR-007	The I/O board shall accept and respond to all ARINC 429 transmissions from the Data Concentrator that contain an SSM of 10b.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

At the bottom, there are buttons for "Save High-Level Requirement Links", "Save Test Case", "Back", and "Undo".

Figure 7-4. Link Test Cases to High-Level Requirements

First choose the Item whose requirements you want to link to. By default, the current Item is selected, but you have the ability to link to requirements of another item. For example, if you have a test that will exercise both the hardware and software of a project, you can link to the appropriate requirements in both items.

The list of Requirements presented changes if you change the item. In the Requirements list area, click the checkbox to the left of one or more High-Level Requirements that the Test Case covers. Clicking the checkbox at the top of the list will select or clear all checkboxes.

Clicking **Save High-Level Requirement Links** will link the Test Case to the selected High-Level Requirement(s) and will take you to a page that views the Test Case including the information about requirements links. Make sure you click this before clicking **Save Test Case** or your links will not be saved.

7.3.7 Exporting Test Cases

Exporting permits you to save the Test Cases to a file that can be 1) cut-and-pasted into the appropriate *<Hardware/Software> Test Cases & Procedures Document* as part of your certification documentation set stored within PACT, or 2) potentially imported into another application.

You can export Test Cases as an HTML page, PDF document, Comma Separated Value (CSV) file or an Excel File (XLS).

To export Test Cases, from the **Test Cases List Page**, click the **Export** button. This will bring up the **Export Test Cases Page** in a new tab.

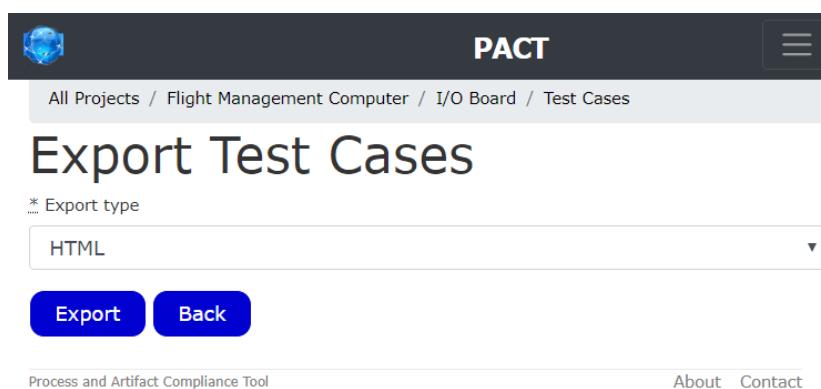


Figure 7-5. Export Test Cases Page

Choose the type of export you would like (HTML, PDF, CSV or XML) and click the **Export** button. Depending on format, the information will either display in this new tab or a file will download to your computer. Close the **Export Test Cases Page** to ensure you do not end up with multiple PACT session tabs.

7.3.8 Importing Test Cases

Importing Test Cases lets you to load the Test Cases for a Hardware/Software Item from another application into the PACT environment. You can import Test Cases from a Comma Separated Value (CSV) file or Excel File (XLS or XLSX).

The file should only contain data (**no headers or extraneous entries**) with each line having the format of:

```
id,caseid,full_id,description,procedure,category,robustness,derived,testmethod,version,item_id,project_id,high_level_requirement_associations,low_level_requirement_associations,derived_justification,created_at,updated_at,organization,archive_id,test_procedure_associations,soft_delete,document_id,model_file_id
```

To import Test Cases, from the **Test Cases List**, click the **Import** button. This brings up the **Import Test Cases Page**.

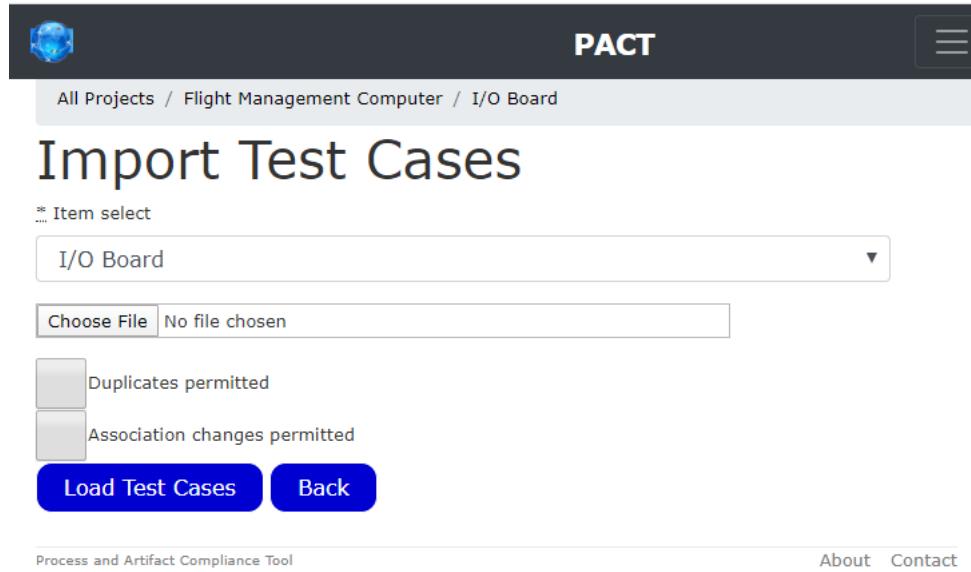


Figure 7-6. Import Test Cases Page

From here, you choose the file including the testcases you would like to import.

Click **Duplicates Permitted** checkbox if you want to permit duplicate IDs to be imported. If you do not check **Duplicates Permitted** and a Test Case already exists with the same ID as in the file when the file is imported, it will cause an error and the file will not be imported.

Click **Association Changes Permitted** if you want to permit changes to Requirement associations. If you do not check **Association Changes Permitted** and a Test Case already exists with the same ID as in the file when the file is imported and it changes the Requirement associations, it will cause an error and the file will not be imported.

Once you have chosen the appropriate options, click to **Load Test Cases** Import the Test Cases into PACT. Click **Back** if you decide not to import.

7.3.9 Renumbering Test Cases

Renumbering Test Cases permits renumbering the test cases starting a given value and incrementing each successive number a specific value. To renumber Test Cases, click **Renumber** from the **Test Cases List**.

All Projects / Requirements Tracing / Software Item / Test Cases

* Start

* Increment

* Leading zeros
3

Renumber **Back**

Process and Artifact Compliance Tool (C)
Airworthiness Certification Services, LLC
 PMB#224
 5500 Olympic Drive
 Suite H-105
 Gig Harbor Wa 98335
 Phone/fax: 425-427-1956

About Privacy and Security Contact

7.4 Managing Test Procedures in PACT

Test Procedures were described in Section 7.1. The following subsections describe how to manage Test Procedures in PACT.

7.4.1 Listing Test Procedures

Clicking the **Test Procedures** link in the **Navigation** column of a specific Hardware/Software Item will display the **Test Procedure List Page**.

All Projects / Flight Management Computer / I/O Board / Test Procedure

Test Procedure List

Field: Value: **Filter**

Search:

Test Procedure ID	Description	File Name	URL	Test Cases	Image	Version					
TP-0001	Runs the linked Test Cases.	Test Procedure 1	Test Procedure 1	I/O_Software-TC-0001 I/O_Software-TC-0002		3					

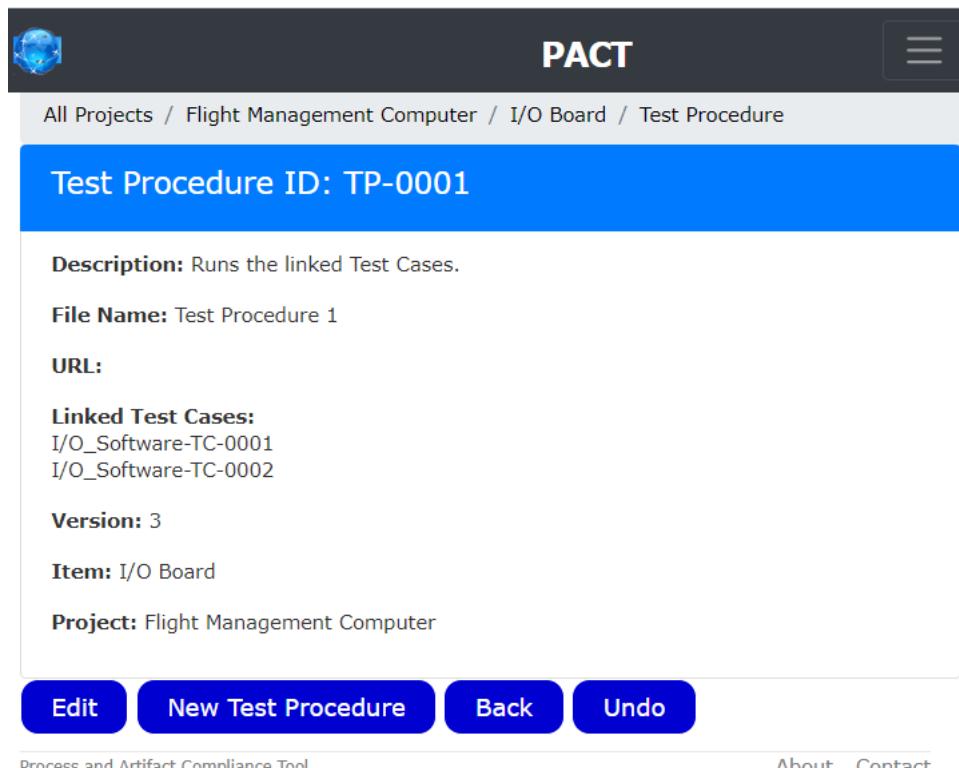
New **Export** **Import** **Renumber** **Back** **Undo**

Figure 7-7. Test Procedure List Page

From here you can view Test Procedures, add a new Test Procedure, edit an existing Test Procedure, mark a Test Procedure as deleted, delete a Test Procedure, link Test Procedures to Test Cases, export Test Procedure, import Test Procedures, or renumber Test Procedures. These activities are described in the following subsections.

7.4.2 Viewing a Test Procedure

To view a Test Procedure, from the **Test Procedure List Page**, click the  icon for a specific Test Procedure. This provides a page showing all the information related to the Test Procedure.



The screenshot shows a web-based application interface for managing test procedures. At the top, there's a dark header bar with the PACT logo on the right. Below it, a navigation bar shows the path: All Projects / Flight Management Computer / I/O Board / Test Procedure. The main content area has a blue header bar with the text "Test Procedure ID: TP-0001". Underneath, there are several data fields with their values:

- Description:** Runs the linked Test Cases.
- File Name:** Test Procedure 1
- URL:** (empty)
- Linked Test Cases:**
 - I/O_Software-TC-0001
 - I/O_Software-TC-0002
- Version:** 3
- Item:** I/O Board
- Project:** Flight Management Computer

At the bottom of the page are four buttons: Edit, New Test Procedure, Back, and Undo. There are also links for Process and Artifact Compliance Tool, About, and Contact.

Figure 7-8. Showing Test Procedure Information

From this Page you can Edit the Test Procedure, add a New Test Procedure or go Back to the **Test Procedures List Page**.

7.4.3 Creating a New Test Procedure

Clicking the  button from the **Test Procedure List Page** brings up the **New Test Procedure Page**.

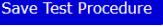
The screenshot displays the 'New Test Procedure' page of the PACT tool. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and Test Procedure. Below the navigation, a breadcrumb trail shows All Projects / Flight Management Computer / I/O Board / Test Procedure. The main title is 'New Test Procedure'. There are several input fields: 'Test Procedure Number' (containing '2'), 'Test Procedure ID' (containing 'TP-0002'), and 'Description' (with a rich text editor toolbar). Below these are fields for 'File name' and 'File Type' (set to 'No Attachment'). A 'Remove image' link is also present. Under 'Item', 'I/O Board' is selected. At the bottom, there are buttons for 'Link Test Case', 'Create Test Procedure', and 'Back'.

Figure 7-9. New Test Procedure Page

Fill in the fields of this page as follows:

- **Test Procedure Number:** This is assigned by the tool and is the next available number.
- **Test Procedure ID:** This required field is filled out by the tool to assign the test case a unique ID. The ID is a combination of the Hardware or Software Item ID plus "TP" plus the next available number. For example, I/O_Software-TP-0001. While you can edit this ID, it is advisable that you do not unless your processes warrant a different naming/numbering scheme. In any case, ensure this field is always unique or PACT will issue an error.
- **Description:** Enter a description for this Test Procedure. This description should include step-by-step instructions for how each test case is to be set up and executed, how the test results are evaluated, and the test environment to be used.
- **File Name:** If you want to link the Test Procedure to a file that executes the procedure, list the file name here.
- **File Type:** If you specify a file name, you then select the type of the file as follows:

- **No Attachment:** If you want to link the Test Procedure information stored in PACT with a file name, but not make the actual connection to this file, select this option.
- **External URL:** You can link the Test Procedure information stored in PACT to an executable file via an external URL with this option. If you choose this Type, a **URL** field appears that you fill in.
- **PACT Documents:** You can link the Test Procedure information stored in PACT to document that is already stored in PACT with this option. If you choose this Type, a **PACT File** option appears that you select from.
- **Upload Image:** You can link the Test Procedure information stored in PACT to a file you want to upload to PACT with this option. If you choose this Type, and **Attachment** field appears that you fill in.
- **Item:** This is set to the current item by default.
- **Link Test Case:** See section 7.4.6.

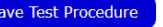
You can Select  to create the Test Procedure. When you save a new Test Procedure, PACT calls this Version 1. Click  to exit without saving the Test Procedure.

7.4.4 Editing a Test Procedure

To edit a Test Procedure, from the **Test Procedure List Page**, click the  icon for a specific Test Procedure or choose the  button from the **Test Procedure List Page**. This brings up the Editing Test Procedure Page. This page is the same as **New Test Procedure Page** (Figure 7-9. New Test Procedure Page) and from here you can edit any of the original information.

When you click the  button, the Test Procedure changes will be saved.



Note that any time you click  -- even if you do not change anything -- increments the version. This allows for tracking a history of the Test Procedure. Also note that this versioning is not the same as the strict versioning that PACT supports for documents that are within the PACT environment.

7.4.5 Deleting Test Procedures

From the **Test Procedure List Page**, Clicking the  or  icons for a specific Test Procedure will delete that Test Case. Before deleting a Test Procedure, a confirmation dialog will appear asking if you are sure you want to delete the Test Procedure. Click **OK** to confirm you want to perform the Delete action or click **Cancel** if you do not want to continue.

The difference between **Mark as Deleted** and **Delete** is that **Mark as Deleted** will leave the Test Procedure but empty its contents replacing it with "deleted." This permits you to maintain a

record that the Test Procedure existed at one time but was deleted. **Delete** will remove any trace of the Test Procedure.

7.4.6 Linking Test Procedures to Test Cases

Test Procedures are associated with Test Cases. (See Section 7.1 for more information on the relationship between Test Cases and Test Procedures). When editing a Test Procedure in PACT, you have the option to link the Test Procedure to one or more Test Cases.

At the bottom of the **New/Edit Test Procedure Page**, you will see the **Link Test Cases** button. Clicking this button will expand the page as follows.

The screenshot shows a user interface for linking test procedures to test cases. At the top, there is a dropdown menu titled "Choose Hardware/Software Item in which to find Test Cases" containing items like CommSoftware, Ethernet_fpga, I/O_Software, and MHW_IO. Below this is a search bar labeled "Search:". A table lists test cases with columns for Hardware/Software Item, Test Case, Description, Test Method, and Robustness. Two entries are shown: I/O_Software-TC-0001 (Description: test ADC function, Test Method: Review/Inspection) and I/O_Software-TC-0002 (Description: CRC test, Test Method: Target). At the bottom, there are buttons for "Save Test Case Links", "Update Test Procedure", "Show", "New Test Procedure", and "Back".

Hardware/Software Item	Test Case	Description	Test Method	Robustness
I/O_Software	I/O_Software-TC-0001	test ADC function	Review/Inspection	<input type="checkbox"/>
I/O_Software	I/O_Software-TC-0002	CRC test	Target	<input type="checkbox"/>

Figure 7-10. Link Test Procedures to Test Cases

First choose the Item whose Test Cases you want to link to. By default, the current Item is selected, but you can link to Test Cases of another item if necessary.

The list of Test Cases presented changes if you change the item. In the Test Case list area, click the checkbox to the left of one or more Test Cases that the Test Procedure applies to. Clicking the checkbox at the top of the list will select or clear all checkboxes.

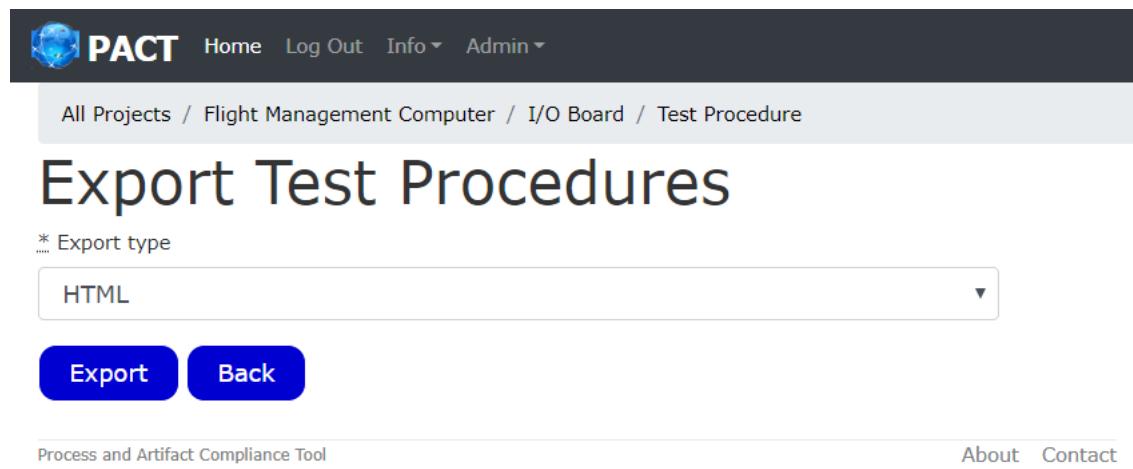
Clicking **Save Test Case Links** will link the Test Procedure to the selected Test Case(s). Make sure you click this before clicking the Create (or Update) Test Procedure button or your links will not be saved.

7.4.7 Exporting Test Procedures

Exporting lets you save the Test Procedures to a file that can be 1) cut-and-pasted into the appropriate *<Hardware/Software> Test Cases & Procedures Document* as part of your certification documentation set stored within PACT, or 2) potentially imported into another application.

You can export Test Procedures as an HTML document, PDF document, Comma Separated Value (CSV) file or an Excel File (XLS).

To export Test Procedures, from the **Test Procedures List**, click the **Export** button. This will bring up the **Export Test Procedures Page** in a new tab.



The screenshot shows the PACT application interface. At the top, there's a dark header bar with the PACT logo, 'Home', 'Log Out', 'Info', and 'Admin' options. Below the header, a breadcrumb navigation bar shows 'All Projects / Flight Management Computer / I/O Board / Test Procedure'. The main content area has a large title 'Export Test Procedures'. Below the title is a form field labeled '* Export type' with a dropdown menu showing 'HTML'. At the bottom of the page are two buttons: 'Export' (highlighted in blue) and 'Back'. At the very bottom, there are links for 'Process and Artifact Compliance Tool', 'About', and 'Contact'.

Figure 7-11. Export Test Procedures Page

As with all other Exports, choose the type of export you would like (HTML, PDF, CSV or XML) and click the **Export** button. Depending on format, the information will either display in this new page or a file will download to your computer. Close the **Export Test Procedures Page** to ensure you do not end up with multiple PACT session tabs.

7.4.8 Importing Test Procedures

Importing Test Procedures lets you load the Test Procedures for a Hardware/Software Item from another application into the PACT environment. You can import Test Procedures from a Comma Separated Value (CSV) file or Excel File (XLS or XLSX).

The file should only contain data (**no headers or extraneous entries**) with each line having the format of:

```
id,procedure_id,full_id,file_name,test_case_associations,url_type,url_description,url_link,version,organization,item_id,project_id,archive_id,created_at,updated_at,description,soft_delete,document_id,file_path,content_type,file_type,revision,draft_version,revision_date,upload_date
```

To import Test Procedures, from the **Test Procedures List**, click the **Import** button. This brings up the **Import Test Procedures Page**.

The screenshot shows the 'Import Test Procedures' page of the PACT application. At the top, there's a navigation bar with the PACT logo and links for Home, Log Out, Info, and Admin. Below that is a breadcrumb trail: All Projects / Flight Management Computer / I/O Board. The main title is 'Import Test Procedures'. A note says '* Item select'. A dropdown menu is open, showing 'I/O Board'. Below it is a file input field with 'Choose File' and 'No file chosen'. There are two checkboxes: one for 'Duplicates permitted' (unchecked) and one for 'Association changes permitted' (unchecked). At the bottom are two buttons: 'Load Test Procedures' (highlighted in blue) and 'Back'.

Process and Artifact Compliance Tool

About Contact

Figure 7-12. Import Test Procedures Page

From here, you choose the file including the test procedures you would like to import.

Click **Duplicates Permitted** checkbox if you want to permit duplicate IDs to be imported. If you do not check **Duplicates Permitted** and a Test Procedure already exists with the same ID as in the file when the file is imported, it will cause an error and the file will not be imported.

Click **Association Changes Permitted** if you want to permit changes to Test Case associations. If you do not check **Association Changes Permitted** and a Test Procedure already exists with the same ID as in the file when the file is imported and it changes the Test Case associations, it will cause an error and the file will not be imported.

7.4.9 Renumbering Test Procedures

Renumbering Test Procedures permits renumbering the Test Procedures starting at one (1) and incrementing each successive number by one (1). To renumber Test Procedures, click

Renumber from the **Test Procedures List Page**.

8 Managing Model Files

Model-Based Design (MBD) is a mathematical and visual method of designing complex control, signal processing and communication systems. MBD is used in many motion control, industrial equipment, aerospace, embedded software, and automotive applications. If you are doing Model Based Design (for example using MATLAB), you will likely want to document and control changes to the Models. This can be done using the Model Files feature provided in PACT.

Model Files are attached to Item requirements and will display with the description. While many Model Files may be attached, when creating or editing requirements you may want to add each Model File Record separately. Within a Hardware/Software Item, you can manage the Model Files shown in the Model File List.

Clicking the Model Files link in the Navigation column of a specific Hardware/Software Item will display the **Model File List Page**.

The screenshot shows the PACT Model File List page. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and Model File. Below that is a breadcrumb trail: All Projects / Flight Management Computer / Communications Board. The main title is "Model File List". There are search and filter fields at the top right. The table has columns: Model File ID, Description, File Name, URL, System Requirements, Requirements, Conceptual Design, and Test Cases. A single row is shown with Model File ID 1, Description "Screen Shot 2020-03-31 at 2.15.15 PM.png", File Name "Screen Shot 2020-03-31 at 2.15.15 PM.png", URL "Screen Shot 2020-03-31 at 2.15.15 PM.png", System Requirements "SYS005", Requirements (empty), Conceptual Design (empty), and Test Cases (empty). To the right of the table are icons for Show, Edit, Mark As Deleted, and Delete. At the bottom are buttons for New, Export, Import, Renumber, and Back, along with links for About and Contact.

Figure 8-1. Model File List Page

This displays a list of Model File records. The term “record” is used here because what you are doing is notifying PACT of the existence of the Model File and providing information about it (see full definition in Appendix B: Glossary of Terms). The record may be pointing to an external URL or a document loaded into PACT. The List contains information about the Model File through its record.

From here you can add a new Model File record, edit an existing Model File, mark a Model File as deleted, delete a Model File, export Model Files, import Model Files. These operations are explained in the following subsections.

8.1 Creating a New Model File Record

From the **Model File List Page**, clicking the **New** button lets you to create a new Model File record and optionally attach a file to it.

The screenshot shows the 'New Model File' page in the PACT application. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Model File. Below that is a breadcrumb trail: All Projects / Flight Management Computer / Communications Board / Model Files. The main title is 'New Model File'. There are several input fields: 'Model File Number' (containing '2'), 'Model File ID' (containing 'MF-0002'), 'Description' (with a rich text editor toolbar showing various icons like bold, italic, underline, etc.), 'File Type' (set to 'No Attachment'), 'Release Model File' (unchecked checkbox), 'Revision' (empty text field), 'Draft Version' (containing '0.1'), 'Revision date' (text field with a date picker icon), 'Item' (dropdown menu set to 'Communications Board'), and a row of buttons for linking System Requirements, Requirements, Conceptual Design, and Test Cases. At the bottom are 'Save Model File' and 'Back' buttons.

Figure 8-2. New Model File Page

Here are the fields you use to create a New Model File record in PACT.

- **Model File ID:** This is prefilled by the tool by the default (i.e., <Item ID>-MF>) or user defined model file prefix by the next available number.
- **Description:** This is a description of the model file.
- **File Type:** File Type lets you attach the model file, in one of three ways: **External URL**,

PACT Document, or **File Upload**. Alternately you can specify **No Attachment**.

- If you want to use an external URL (like from GitHub choose External URL). A new **URL** field will appear below the **File Type** field. Paste or type the URL in the field and that will attach the model file via the external URL.
- On the other hand, if you already added the model file files in PACT, you can choose **PACT Documents**. This causes PACT File field to appear where you select the file from the list of Documents within PACT.
- If you would rather not use an External URL or first upload the files in PACT, you can choose **File Upload**. The **Attach File** field will appear. Click **Choose File** and navigate to the file you want to upload.
- **Release Model File**: If you are changing a model file and you are ready to indicate it is a formal Release (i.e., it has been reviewed and approved), click this button.
- **Revision**: You may enter the revision of the Model File here.
- **Draft Version**: This will be prefilled by the tool as per the versioning information described in Section 5.2.
- **Revision Date**: This is the date of the revision.



Note: The linking capability on this form is explained in Section 8.4.

After you fill in all the pertinent fields, you can Select **Save Model File** to create the Model File Record or **Back** to exit without saving the Model File.

8.2 Editing a Model File Record

To edit a Model File record, from the **Model File List Page**, click the  icon for a specific Model File or choose the **Edit** button from the **Model File View Page**. Editing a Model File record is identical to creating a new Model File record and all the fields are the same.

 *Note that any time you click **Save Model File** -- even if you do not change anything – PACT increments the version. This allows for tracking a history of the model file record. Also note that this versioning is not the same as the strict versioning that PACT supports for documents that are within the PACT environment.*

8.3 Deleting a Model File

From the **Model File List Page**, clicking the  or  icons for a specific Model File record will delete that Model File record. Before deleting a Model File record, a confirmation dialog will

appear asking if you are sure you want to delete the Model File. You have to select **OK** to complete the deletion.

The difference between **Mark as Deleted** and **Delete** is that **Mark as Deleted** will leave the model file record but will “strikeout” the contents. This permits you to maintain a record that the Model File existed at one time but was deleted. **Delete** will remove any trace of the Model File. You should use the **Delete** capability with caution.

8.4 Linking a Model File to Requirements

When adding or editing a Model File record, you have the option to link the Model File to one or more System, High-Level/Low-Level Requirements or Test Cases. You will see these buttons at the bottom of the **Edit Model File Page**. Linking to Requirements for Model Files is the same as for all other requirements. Please see Section 4.3.5 for examples of how to link to other requirements.

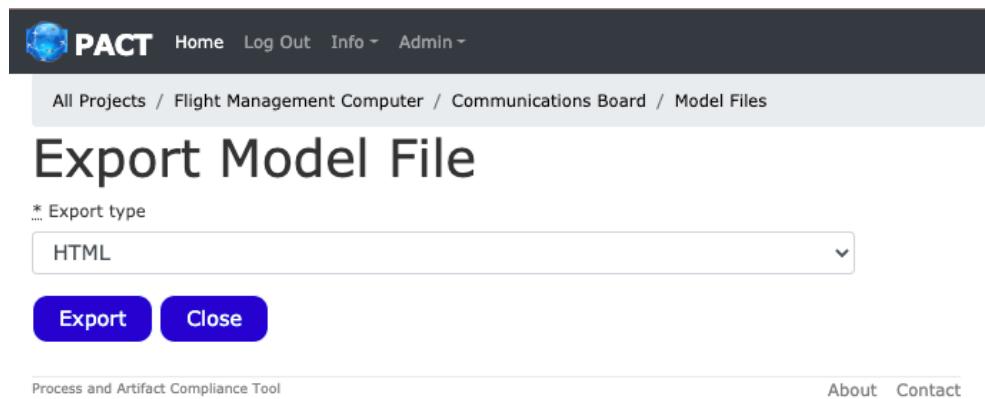
To view a Model File record from the **Model File List Page**, click the  icon for a specific Model File record or choose the  button from the **Model File Edit Page**.

8.5 Exporting a Model File

Exporting a Model File permits you to save the model file to a file that can be 1) cut-and-pasted into the appropriate *<Hardware/Software> Design Document* as part of your certification documentation set stored within PACT, or 2) potentially imported into another application.

You can export Model File as an HTML document, PDF document, Comma Separated Value (CSV) file or an Excel File (XLS).

To export Model File, from the **Model File List Page**, click the  button. This opens the **Export Model File Page** (in a new browser tab).



The screenshot shows the 'Export Model File' page. At the top, there's a navigation bar with the PACT logo, 'Home', 'Log Out', 'Info', and 'Admin'. Below that is a breadcrumb trail: 'All Projects / Flight Management Computer / Communications Board / Model Files'. The main title is 'Export Model File'. Underneath, there's a label 'Export type' followed by a dropdown menu set to 'HTML'. At the bottom are two buttons: 'Export' (highlighted in blue) and 'Close'. The footer contains the text 'Process and Artifact Compliance Tool' and links for 'About' and 'Contact'.

Figure 8-3. Export Model File Page

Choose the type of export you would like (HTML, PDF, CSV or XML) and click the **Export** button. Depending on format, the information will either display in this new page or a file will download to your computer. Close the **Export Model File Page** to ensure you do not end up with multiple PACT session tabs.

8.6 Importing a Model File

Importing a Model File permits you to load the Model File for a Hardware/Software Item from another application. You can import Model File from a Comma Separated Value (CSV) file or Excel File (XLS or XLSX).

The file should only contain data (**no headers or extraneous entries**) with lines in the format as follows (on one line):

```
id,model_id,full_id,description,file_path,file_type,url_type,url_link,  
url_description,soft_delete14lerivedd,derived_justification,system_req  
uirement_associations,high_level_requirement_associations,low_level_re  
quirement_associations,test_case_associations,version,revision,draft_v  
ersion,revision_date,organization,project_id,item_id,archive_id,create  
d_at,updated_at,upload_date
```

To import a Model File, from the **Model File List Page**, click the **Import** button.

The screenshot shows the 'Import Model File' page of the PACT application. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and a dropdown menu. Below the navigation bar, the current location is shown as All Projects / Flight Management Computer / Communications Board / Model Files. The main title is 'Import Model File'. There is a dropdown menu labeled 'Item select' with 'Communications Board' selected. Below it is a 'Choose File' input field showing 'No file chosen'. There are two checkboxes: 'Duplicates permitted' and 'Association changes permitted'. At the bottom are two buttons: 'Load Model Files' (highlighted in blue) and 'Back'.

Figure 8-4. Import Model File Page

To Import a Model File, perform the following tasks:

- 1 Choose the **File** you would like to import (in the format shown previously).

- 2** Click **Duplicates Permitted** if you want to permit duplicate model file IDs to be imported. If you do not check Duplicates Permitted and a Model File record already exists with the same ID as in the file when the file is imported, it will cause an error and the file will not be imported.
- 3** Click **Association Changes Permitted** if you want to permit changes to Low-Level Requirement or High-Level Requirement associations. If you do not check this and a Model File record already exists with the same ID as in the file when the file is imported and it changes the Low or High-Level Requirement associations, it will cause an error and the file will not be imported.

9 Establishing Requirements Traceability

Safety-critical programs are requirements-driven programs. In the initial stages of the design life cycle, your team writes good requirements that define the design's functionality. From these, a development team creates the design, linking the design itself to the corresponding requirements to ensure the design implements all of the required functionality. Concurrently (and usually independently), a verification team writes test cases from the requirements to ensure the implemented design performs the functions as defined by the requirements.

To ensure all the functionality is implemented and tested, you need to link the design implementation and testing artifacts to the requirements. This is called requirements traceability. The full set of linkages between requirements and their corresponding design implementation and/or tests (and results) is called the Requirements Traceability Matrix (RTM). PACT can assist with producing an RTM.



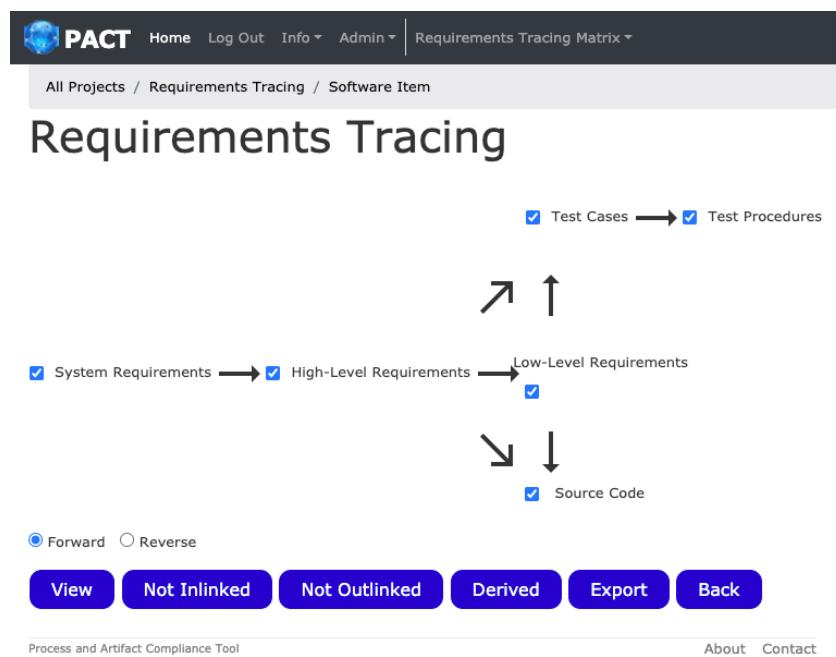
Note: The documents that capture the RTM may be called a Software Traceability Matrix (STM) or Hardware Traceability Matrix (HTM) for DO-178C or DO-254 programs, respectively.

In order to create the Requirements Traceability Matrix, you have to have previously established the links between the following objects as described in these sections:

- System Requirements to Software High-Level Requirements (or Hardware Requirements). See Section 4.3.5.
- Software High-Level Requirements to Low-Level Requirements (or Hardware Requirements to Conceptual Design). See Section 4.3.12.
- Software High- or Low-Level Requirements to Source Code (or Hardware Requirements or Conceptual Design to Source Code). See Section 0.
- Software High- or Low-Level Requirements to Test Cases (or Hardware Requirements or Conceptual Design to Test Cases). See Section 7.3.6.
- Software/Hardware Test Procedures to Test Cases. See Section 7.4.6.

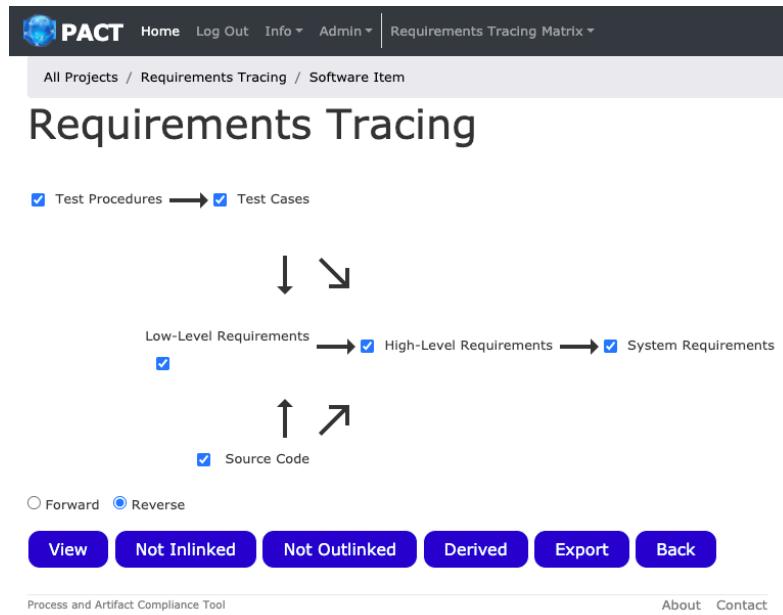
9.1 Requirements Traceability Matrix

Clicking the **RTM** link in the **Navigation** Column of a specific Hardware/Software Item will display the **Requirements Tracing Page**.

*Figure 9-1. Requirements Tracing Page*

This page gives you options to View or Export various levels of and options for requirements tracing throughout the project.

Each type of requirement has a checkbox. When clicked, that particular requirement is included in the Requirements Trace. The flow of the trace can be from higher level requirements to lower level requirements (i.e., **Forward**; this is the default) or from lower level requirements to higher level requirements (i.e., **Reverse**).

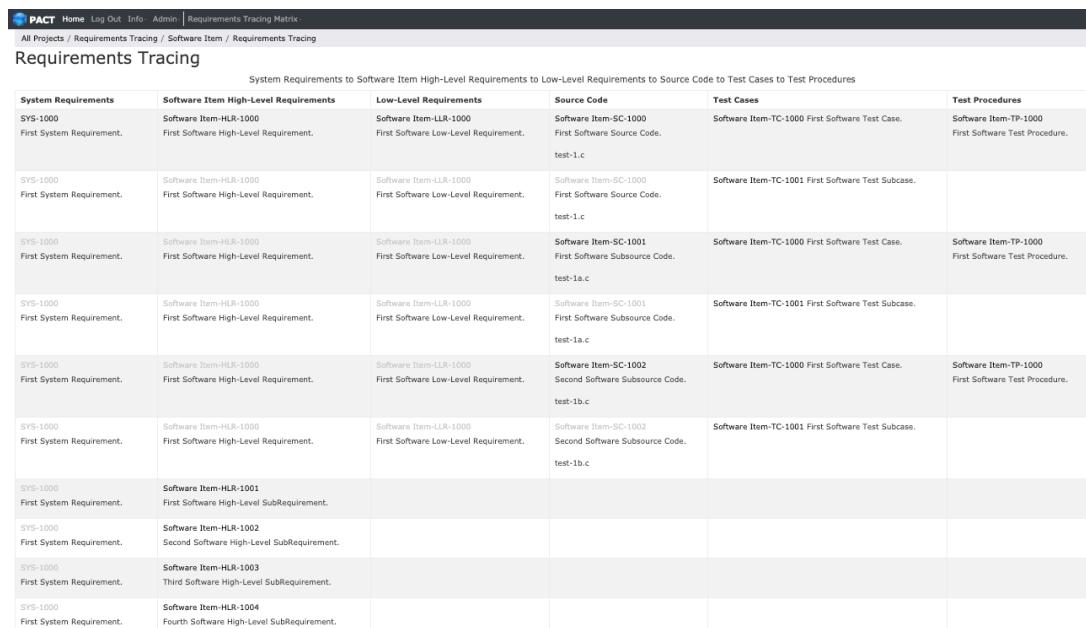
*Figure 9-2. Reversed Requirements Tracing Page*

Not all combinations of checkboxes are possible. For example, you could not trace from System Requirements to Test Procedures without including HLRs and Test Cases. PACT will not let you create an invalid combination.

You can view the RTM or select an Export format such that the RTM can be copied into documentation or imported into another tool.

9.2 Viewing the RTM

When you have chosen the requirements that you want to include in your trace, you can click the **View** button to see the results of the trace.



System Requirements to Software Item High-Level Requirements to Low-Level Requirements to Source Code to Test Cases to Test Procedures					
System Requirements	Software Item High-Level Requirements	Low-Level Requirements	Source Code	Test Cases	Test Procedures
SYS-1000 First System Requirement.	Software Item-HLR-1000 First Software High-Level Requirement.	Software Item-LLR-1000 First Software Low-Level Requirement.	Software Item-SC-1000 First Software Source Code. test-1.c	Software Item-TC-1000 First Software Test Case.	Software Item-TP-1000 First Software Test Procedure.
SYS-1000 First System Requirement.	Software Item-HLR-1000 First Software High-Level Requirement.	Software Item-LLR-1000 First Software Low-Level Requirement.	Software Item-SC-1000 First Software Source Code. test-1.c	Software Item-TC-1001 First Software Test Subcase.	
SYS-1000 First System Requirement.	Software Item-HLR-1000 First Software High-Level Requirement.	Software Item-LLR-1000 First Software Low-Level Requirement.	Software Item-SC-1001 First Software Subsource Code. test-1a.c	Software Item-TC-1000 First Software Test Case.	Software Item-TP-1000 First Software Test Procedure.
SYS-1000 First System Requirement.	Software Item-HLR-1000 First Software High-Level Requirement.	Software Item-LLR-1000 First Software Low-Level Requirement.	Software Item-SC-1001 First Software Subsource Code. test-1a.c	Software Item-TC-1001 First Software Test Subcase.	
SYS-1000 First System Requirement.	Software Item-HLR-1000 First Software High-Level Requirement.	Software Item-LLR-1000 First Software Low-Level Requirement.	Software Item-SC-1002 Second Software Subsource Code. test-1b.c	Software Item-TC-1000 First Software Test Case.	Software Item-TP-1000 First Software Test Procedure.
SYS-1000 First System Requirement.	Software Item-HLR-1000 First Software High-Level Requirement.	Software Item-LLR-1000 First Software Low-Level Requirement.	Software Item-SC-1002 Second Software Subsource Code. test-1b.c	Software Item-TC-1001 First Software Test Subcase.	
SYS-1000 First System Requirement.	Software Item-HLR-1001 First Software High-Level SubRequirement.				
SYS-1000 First System Requirement.	Software Item-HLR-1002 Second Software High-Level SubRequirement.				
SYS-1000 First System Requirement.	Software Item-HLR-1003 Third Software High-Level SubRequirement.				
SYS-1000 First System Requirement.	Software Item-HLR-1004 Fourth Software High-Level SubRequirement.				

Figure 9-3. Requirements Tracing Example

If you select **View**, PACT opens a new tab displaying the RTM. Close that tab or window when you are finished viewing the content.



Note that this traceability view can get very wide and may be hard to read in the PACT User Interface itself. Exporting the RTM to a different format is an alternate solution to viewing the RTM.

9.3 Exporting the RTM

Exporting permits you to save the RTM to a file that can be 1) reviewed outside of PACT, and/or 2) cut-and-pasted into the appropriate document as part of your certification documentation set stored within PACT.

If you select **Export** from the **Requirements Tracing Page**, PACT opens the **Export Requirements Tracing Matrix Page** in a new tab.

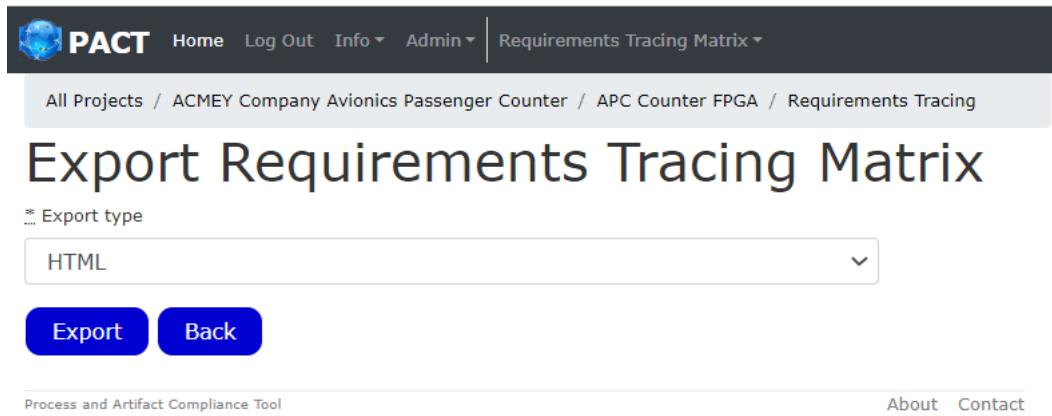


Figure 9-4. Export Requirements Tracing Matrix Page

As with all the other types of Export operations, depending on chosen format, the information will either display in this new page or a file will download to your computer. Close the **Requirements Tracing Matrix Page** to ensure you do not end up with multiple PACT session tabs.

10 Holding Reviews

Reviews are a form of verification that is very common in safety-critical programs. Reviews provide verification that documents (and/or artifacts) have been developed and processes have been performed as planned.

10.1 Understanding Reviews

Reviews allow team members to convene (physically or virtually) to examine and discuss a document, artifact, or process to ensure compliance to plans and standards. A typical DO-254 or DO-178C process will involve numerous reviews broken down into two categories:

1. Peer Reviews: These reviews focus on examining a document or other artifact that is the output of a particular process or activity to ensure it is complete and adheres to plans and/or standards.
2. Transition Reviews: These reviews focus on reviewing a completed development life cycle phase to ensure everything has been accomplished that needs to be prior to transitioning into the next phase.

Figure 10-1 provides an example of what these reviews may look like in the context of a DO-254 program.

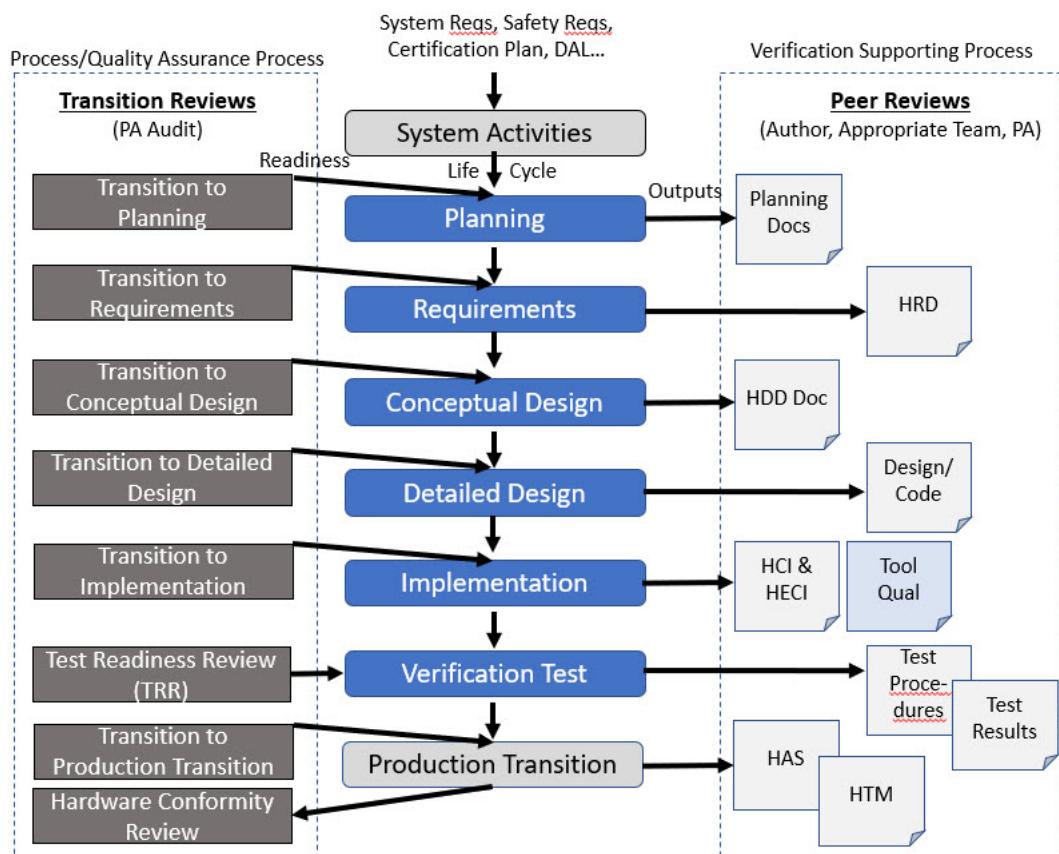


Figure 10-1. Typical DO-254 Program Reviews

Similarly, Figure 10-2 provides an example of what these reviews may look like in the context of a DO-178C program.

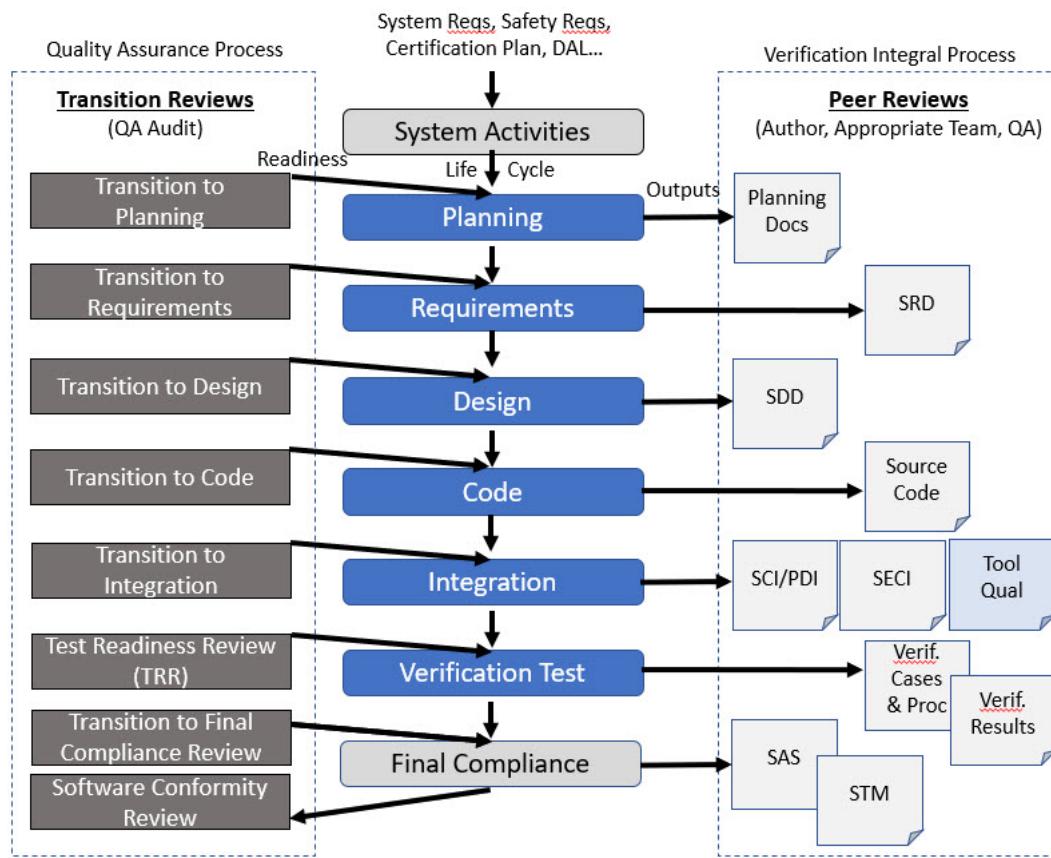


Figure 10-2. Typical DO-178C Program Reviews

PACT directly supports the review processes as shown in these figures for DO-254 and DO-178C programs. In addition, PACT has sufficient flexibility to support different types of reviews for other safety-critical programs.

10.2 PACT Review Flow

PACT provides two distinct facets of reviews:

- **Project perspective:** This feature provides status for all Hardware/Software Items within a Project. See Section 10.3 for more information.
- **Hardware/Software Item perspective:** This feature involves holding the actual reviews for each Item being developed within a Project. This is described in the figure and explanation that follow.

Figure 10-3 provides a high-level perspective of the Review flow within the PACT tool. The steps are described in summary in this section and in detail in the referenced subsections.

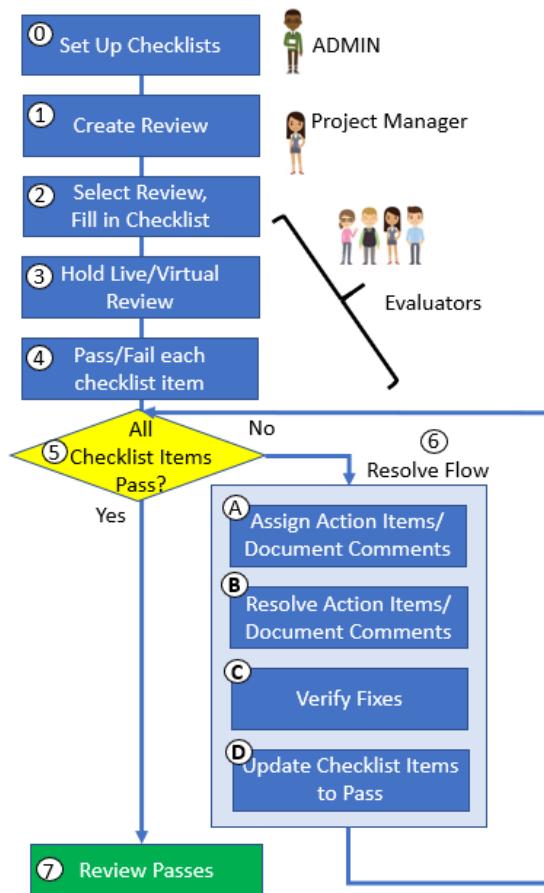


Figure 10-3. Review Flow in PACT

These are the basic steps involved in holding a review in PACT.

Prior to the actual review flow, you should ensure that your Administrator and/or Project Manager has set up your Review Checklist Templates. See Section 10.5 for details.

- 1 **Create Review:** The person who is responsible for setting up the review (usually the Project Manager) creates a New Review. See Section 10.6 for details.
- 2 **Select Review/Fill in Checklist:** Each person assigned to be an Evaluator for a review gets an email notice that invites them to the review. They go to the review, sign in, and fill in the Checklist when directed.
- 3 **Hold Live/Virtual Review:** The team of evaluators meets live or virtually, as a group or individually, to review the document or artifact, and go through each question in the checklist one at a time. See Section 10.8 for details.
- 4 **Pass/Fail Each Checklist Item:** Each evaluator sets a status for each checklist question to Pass, Fail or NA. PACT keeps tabs on how each evaluator sets the status for each question and can provide a Consolidated Checklist view. See Section 10.8.2 for details.
- 5 **All Checklist Items Pass?:** If all evaluators set all questions to Pass, the review is complete.

6 Resolve Items: When there are one or more questions with a Fail status, this means those issues need to be identified, tracked and resolved as described below. See Section 10.9 for details.

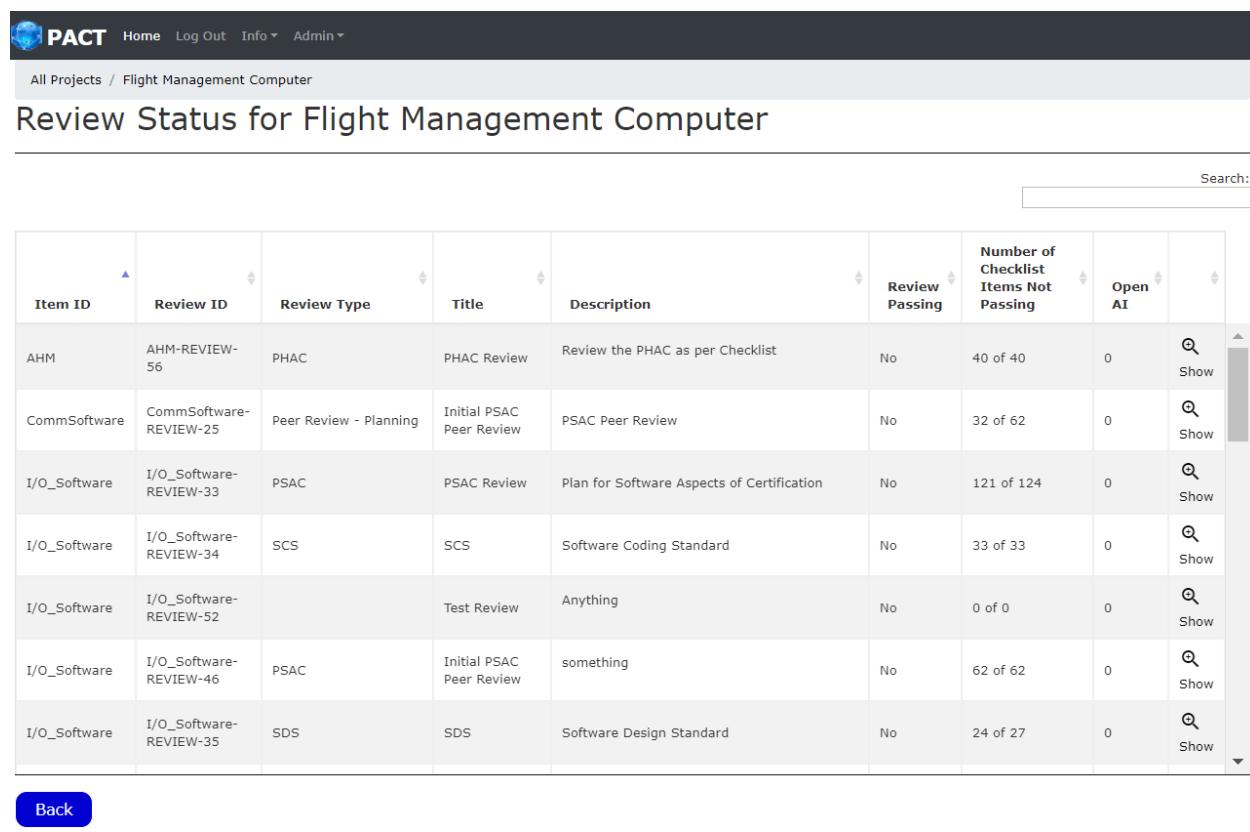
- A Assign Action Items/Document Comments: Issues found during a review need to be tracked to resolution. For Peer Reviews, they can be tracked by Action Items or in Document Comments. In Transition Reviews, issues must be tracked by Action Items. In either case, the issue should be assigned to a team member for resolution.
- B Resolve Action Items/Document Comments: If you are assigned an Action Item or Document comment to address, you resolve it and then assign a Verifier to ensure the issue is truly resolved.
- C Verify Fixes: If you are assigned to Verify that an issue has been resolved, you do so and then report back to the team that the issue is indeed fixed.
- D Update Checklist Items to “Pass”: The review evaluators then go back into the Checklist for the review and edit it to change each fixed item from Fail to Pass. Once all Evaluators set all checklist items pass, go to #7.

7 Review Passes: See Section 10.10 for details.

10.3 Project Review Status

At the Project level, you can view the Review status for all the Hardware/Software Items that comprise the system-level Project. This will give you a good idea of your project status. As a Project progresses, teams review documents, artifacts, and life cycle processes. Reviews yield issues that the team resolves, and through an iterative review-fix process, eventually the review passes, indicating that the document/artifact or life cycle phase has been completed and approved.

You can view the review status for all Items that comprise the system Project by clicking **Review Status** in the **Navigation** column for the Project of interest. This brings up the **Review Status Page** for that Project.



The screenshot shows a web-based application titled "Review Status for Flight Management Computer". At the top, there's a navigation bar with links for Home, Log Out, Info, and Admin. Below the navigation, a breadcrumb trail shows "All Projects / Flight Management Computer". The main content area has a search bar labeled "Search:".

Item ID	Review ID	Review Type	Title	Description	Review Passing	Number of Checklist Items Not Passing	Open AI	
AHM	AHM-REVIEW-56	PHAC	PHAC Review	Review the PHAC as per Checklist	No	40 of 40	0	 Show
CommSoftware	CommSoftware-REVIEW-25	Peer Review - Planning	Initial PSAC Peer Review	PSAC Peer Review	No	32 of 62	0	 Show
I/O_Software	I/O_Software-REVIEW-33	PSAC	PSAC Review	Plan for Software Aspects of Certification	No	121 of 124	0	 Show
I/O_Software	I/O_Software-REVIEW-34	SCS	SCS	Software Coding Standard	No	33 of 33	0	 Show
I/O_Software	I/O_Software-REVIEW-52		Test Review	Anything	No	0 of 0	0	 Show
I/O_Software	I/O_Software-REVIEW-46	PSAC	Initial PSAC Peer Review	something	No	62 of 62	0	 Show
I/O_Software	I/O_Software-REVIEW-35	SDS	SDS	Software Design Standard	No	24 of 27	0	 Show

[Back](#)*Figure 10-4. Project-Level Review Status Page*

From the **Review Status Page**, clicking the  icon for a specific review will display the information for that specific review.

At the Project level, you cannot make any changes to the review content. It is a view-only perspective intended to provide status information for the Project as a whole.

All Projects / Flight Management Computer / I/O Board / Transition to Requirements

Review ID: I/O_Software-REVIEW-59

Details:

Title: Transition to Requirements

Review Type: Software Transition to Requirements Phase

Review created by: Michelle Lange

Problem Reports Addressed:

FMC-PR-4: A new problem

Evaluators: Michelle Lange , Paul Carrick , Michelle Lange*

Project Managers: Michelle Lange

Configuration Managers: Michelle Lange

Quality Assurance: Michelle Lange*

Team Members: Admin User , Dave Newton , Michelle Lange , Paul Carrick , Paul J. Carrick* , Steve Gregor , Tammy Reeve

Airworthiness Certification Representatives: Tammy Reeve

Review Date: 2020-04-29

Description:

This review will ensure we've done everything needed in Planning and are ready to go to the Requirements phase.

Version: 2

Item: I/O Board

Project: Flight Management Computer

Action Items: Link

Open Action Items: 1

Checklist Items:

Checklist Items Per Checklist:	18
Evaluators:	3
Total Assigned Checklist Items:	54
Passing Checklist Items:	4
Failing Checklist Items:	1
Checklist Items Listed as N/A:	0
Incomplete Checklist Items :	49
Percentage Complete:	9.26

Figure 10-5. Specific Review Status Page

This page shows status for the specific review. At the bottom of the specific **Review Status Page** is the Checklist, which contains all the questions asked in the review.

Checklist:

Search:

ID	Description	Applicable DAL	Reference	Supplements	Evaluator	Status	Notes
1	Has the Plan for Software Aspects of Certification (PSAC) been produced, reviewed, released and approved (with all action items closed)?	B	11.1; Table A-1.1-7	Model Based Object Oriented Formal Method	Michelle Lange	Fail	Problem on PSAC page 6
2	Has the Software Development Plan (SDP) been produced, reviewed, released and approved (with all action items closed)?	B	11.2; Table A-1.1-7	Model Based Object Oriented Formal Method	Michelle Lange	Pass	
3	Has the Software Verification Plan (SVP) been produced, reviewed, released and approved (with all action items closed)?	B	11.3; Table A-1.1-7	Model Based Object Oriented Formal Method	Michelle Lange	Pass	
4	Has the Software Configuration Management Plan (SCMP) been produced, reviewed, released and approved (with all action items closed)?	B	11.4; Table A-1.1-7	Model Based Object Oriented Formal Method	Michelle Lange	Pass	
5	Has the Software Quality Assurance Plan (SQAP) been produced, reviewed, released and approved (with all action items closed)?	B	11.5; Table A-1.1-7	Model Based Object Oriented Formal Method	Michelle Lange	Pass	
6	Has the Software Requirements Standards document (SRS) been produced, reviewed, released and approved (with all action items closed)?	B	11.6; Table A-1.5	Model Based Object Oriented Formal Method	Michelle Lange		
Use the Software Design Standards document (CASC) been produced							

[Edit Review](#) [New Review](#) [Consolidated Checklist](#) [Sign-In Sheet](#) [Back](#) [Undo](#)

Figure 10-6. Checklist Associated with a Review

PACT has a predefined set of Checklists that are used to drive the reviews for DO-254 and DO-178C programs. Using these predefined Checklists can dramatically facilitate the review process. For more information on Checklists, see Section 10.5.

10.4 The Review Process for Hardware/Software Items

To access the Review functionality for a specific Hardware/Software Item, click **Hardware/Software Items** for a specific Project from the **Projects List Page**. For a specific Item, in the **Navigation** column, select **Reviews**. This brings up the **Reviews List Page** for that specific Item.

The screenshot shows the 'Reviews List' page of the PACT application. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and Reviews. Below the navigation bar, a breadcrumb trail shows 'All Projects / Flight Management Computer / A Hardware Module'. The main title 'Reviews List' is displayed prominently. To the right of the title is a search input field labeled 'Search:' with a placeholder 'Search reviews...'. Below the title is a table with the following columns: Review ID, Specific Review Type, Title, Evaluation Date, Description, Review Passing, Action Items, and Version. There is one row in the table representing a review for 'PHAC Review' dated '2020-04-13'. The 'Action Items' column contains the text: '1 open AI(s)
You must take action regarding non-compliance issue #2 on PHAC review.' The 'Review Passing' column shows 'No'. The 'Version' column shows '1'. To the right of the table are three buttons: 'Show' (with a magnifying glass icon), 'Edit' (with a pencil icon), and 'Delete' (with a trash bin icon). At the bottom left of the page are two buttons: 'New Review' and 'Back'. At the bottom right are links for 'About', 'Privacy and Security', and 'Contact'. On the left side of the page, there is a sidebar with contact information: Process and Artifact Compliance Tool (C) Airworthiness Certification Services, LLC PMB#224 5500 Olympic Drive Suite H-105 Gig Harbor Wa 98335 Phone/fax: 425-427-1956.

Review ID	Specific Review Type	Title	Evaluation Date	Description	Review Passing	Action Items	Version			
56	PHAC	PHAC Review	2020-04-13	Review the PHAC as per Checklist	No	1 open AI(s) You must take action regarding non-compliance issue #2 on PHAC review.	1	Show	Edit	Delete

Figure 10-7. Item-Level Reviews List Page

Here you can scroll through all the reviews previously held and/or in process on the current Hardware or Software Item.

The next subsections describe the review process shown in Figure 10-3 in more detail.

10.5 Setting Up Checklists

ACS provides a set of checklist templates that are used within the scope of reviews. They can be used as-is or modified.

Checklists are used in Reviews to ensure that all compliance requirements have been fulfilled. One or more Evaluators (i.e., Reviewers) will fill out each Checklist. When all Checklists have passed, the Review will pass.

PACT supports several levels of checklists: ACS (i.e., Global) Checklist Templates, Organizational Checklist set, Review Master Checklists, and Individual Assigned Checklists.

ACS maintains the Global Checklist Templates and like Document Templates, these are copied into the Organizational Checklist set by the Administrator. The Project Manager and/or Configuration Manager can modify or add Checklist Templates at the Organizational level. This was described in Section 5.3 in the context of Document Templates, but the same process holds true for setting up Checklist Templates.

When a Review is created, the appropriate Organizational Checklist is copied into the Master Checklist based on the Hardware/Software Item's Category and DAL, along with the specific type of Review.

As Evaluators are assigned to the Review, a copy of the Master Checklist is copied into the User's (Assigned) Checklist. The User then fills out their assigned Checklist.

When the individual users' checklists are consolidated, PACT creates a Consolidated Checklist for the Review.

10.6 Setting Up a New Review

From the **Reviews List Page**, you can click on the  **New Review** button to set up a new Review.



For Peer Reviews, the person setting up the review is typically the document Author.

For Transition Reviews, the person setting up the review is typically the Quality/Process engineer.

Clicking the  **New Review** button brings up the **New Review Page**.

New Review

* Review ID
65

* Title
SDP Review

Description

This review is to evaluate the SDP to ensure it captures all the correct information.

Review type
Peer Review

Specific Review
Software Development Plan (DO-178 Peer Review)

Problem reports addressed

FMC-PR-1: Issue with software signal IN_2
FMC-PR-2: REQ-002 not clear
FMC-PR-4: A new problem

File Under Review

Attachment Type
PACT Documents

PACT Documents

S_02_C_SD.P DOWNLOAD.r2.doc

Additional Reference Material

Type File/URL

File Type
PACT Documents

S_01_C_PSAC.DOWNLOAD.r2.doc

Add file

Unassigned Users
Michelle Lange
Paul Carrick
Dave Newton
Steve Gregor

Evaluators
Michelle Lange
Paul Jeffrey Carrick
Tammy Reeve

Review Date
05/08/2020

* item
I/O Board

Create Review Back

Figure 10-8. New Review Page

To set up a new Review, fill in this page as follows:

- **Review ID:** This is set by the tool itself to keep track of the Review. The Review is assigned the next available number.
- **Title:** Provide a descriptive name for the Review.
- **Description:** Provide as much detail as you can about the review and its purpose.
- **Review Type:** Choose between Peer Review or Transition Review, or Other.

- **Specific Review:** Select the specific review from the list of Reviews (which are shown based on the Review Type chosen). Note that this selection then assigns the appropriate checklist. See Section 10.5 for more information on how checklists can be set up for your Project.
- **Problem reports addressed:** In some cases, a Review may involve examining one or more related PRs. If this is the case, select the related PR(s) here.
- **File under Review:** Select the document or artifact that is being reviewed. This can point to an external URL, a PACT document, or you can upload a new file into PACT. This is generally required for a Peer review, which reviews a document or artifact.
- **Additional Reference Material:** If there are other documents that may be good reference for the review, select them here.
- **Evaluators:** Select the Team Members from the left-hand box and click the  button to make these people Evaluators in this review. An Evaluator is a participant in the review.
- **Review Date:** The review date is set to the current date, but you can override this if the Review is being held on a different day.
- **Item:** This indicates what item this review applies to.

Click  to set up the new Review.

PACT sends an email to each Evaluator to let them know they have been invited to a review.

A Checklist has been assigned to you.

 info@airworthinesscert.com <info@airworthinesscert.com> Tuesday, March 10, 2020 at 10:10 AM
To: Paul Carrick

Dear Paul Carrick:

You have been assigned to be an Evaluator in a New Review. Please see below for the details.

Review Detail:

ID: 9
Title: HDP
Description: Hardware Development Plan
Type:
Evaluation Date: 2020-01-07
Evaluators: Dave Newton , Steve Gregor , Paul Carrick

PACT Automation
pact@airworthinesscert.com

Figure 10-9. Email to Evaluators



Note that if you do not receive such an email, it may be because the “Notify on Changes” option may be turned off in your User Profile.

10.7 Joining a New Review

If you are selected as an Evaluator, this means you need to join a new Review. You do this by clicking the link in your notification email. Alternately, navigate to the pertinent Hardware/Software Item for your Project and then click on **Reviews**. This will open the **Reviews List Page**.

Scroll through the list of reviews listed and click the button for the Review that has the Review ID number that matches what is shown in the email you received.

This brings up the **Editing Review Page**.

All Projects / ACMEY Company Avionics Passenger Counter / _Avionics Passenger Counter (APC) Controller Board

Editing Review

Review ID: APC_CB-REVIEW-1

* Review ID
1

Specific Review

* Title
Review of PHAC draft

Unassigned Users

Admin User
Michelle Lange*
Tammy Reeve*
Steve Gregor*

Evaluators

Dave Newton
Michelle Lange

Review Date
06/25/2020

Description

Formal review of the Plan for Hardware Aspects of Certification (PHAC) 800-PHAC-01 rev 0.1.
Exit criteria for the review is to create the initial baseline of the PHAC at Revision A.

Problem reports addressed

APC-PR-1: PHAC PR1
APC-PR-2: Missing Safety Concerns
APC-PR-3: Planning Process Issues

* Item

Figure 10-10. Information Portion of Editing Review Page

The top part of the page provides information about the Review and includes all the fields in the **New Review Page** (Figure 10-8). Scroll to the bottom of this page. You will see the **Checklist** that applies to the specific Review.

Checklist:

Fill in Checklist

Search:

ID	Description	Applicable DAL	Reference	Supplements	Evaluator	Status	Notes
1	Has the PHAC been subjected to appropriate change and configuration control (HC1)?	A	Table A-1		Michelle Lange	Pass	
2	Does the PHAC list the tools being used as part of the Life Cycle process and describe or reference the Tool Qualification Plan? (See TQP checklist for more information)	A	11.4; AC20-152A		Michelle Lange	Pass	
3	Does the PHAC address how to address open problem reports?	A	AC20-152A		Michelle Lange	Pass	
4	Does the PHAC address supplier oversight?	A	4.2.9		Michelle Lange	Pass	
5	Does the PHAC include a Certification Schedule (including major program milestones and the dates when hardware design life cycle data will be submitted to the certification authority)?	A	10.1.1-8		Michelle Lange	Pass	
6	Does the PHAC include Alternative Methods (if any are proposed)?	A	10.1.1-7		Michelle Lange	Pass	
7	Does the PHAC include design assurance considerations for Level A or B functions as described in Appendix B?	A	10.1.1-6		Michelle Lange	Pass	
8	If PDH is to be used, does the PHAC detail the plan for compliance for this as per AC20-152A? (See separate checklist for this topic)	A	AC20-152A CD-1:11		Michelle Lange	Pass	
9	If COTS devices are to be used, does the PHAC detail the plan for compliance for these devices as per AC20-152A? (See separate checklist for this topic)	A	AC20-152A COTS-1:8		Michelle Lange	Pass	
10	If COTS IP is to be used, does the PHAC detail the plan for compliance for this COTS IP as per AC20-152A? (See separate checklist for this topic)	A	AC20-152A IP-1:6		Michelle Lange	Pass	

Save Changes **Show** **Sign-In Sheet** **Back**

Figure 10-11. Checklist Portion of Editing Review Page

During the review, you will be expected to review every checklist item (i.e., question) on this checklist and determine if the document/artifact (or process) under review passes or fails each question.

10.8 Participating in a Review

How each team chooses to handle the review process should be documented in the planning documents (e.g., the Quality Assurance Plan). A review may be a physical gathering, a live/online conference, or even a period of independent review by each of the evaluators. How your team chooses to hold reviews may affect the review process of Figure 10-3 to some extent.

For example, your plans may indicate that each Evaluator will review all checklist questions independently and then a meeting will be held to discuss the review results. Alternately, your plans may require a physical (or online) Review meeting to work through the checklist items all together as a team. You may have a meeting after all items have been resolved to ensure

resolution, or this may be done by individuals assigned to be verifiers of the issues. PACT supports this flexibility and Figure 10-3 just provides a typical flow to explain the steps and activities that will be performed.

Regardless of how it is held, each Evaluator will need to Sign Into the Review, fill out the Checklist, capture Action Items or Document Comments as needed (and track them to closure), and then revisit the Checklist when all the issues have been addressed and verified such that the document, artifact, or process under review can ultimately pass. Each of these steps are discussed further in the following subsections.

10.8.1 Signing into the Review

If you are an evaluator, participating in a Review in the PACT environment involves going into the **Editing Review Page** and signing into the review. You can sign-in prior to filling out the checklist or after completing the checklist, depending on your process.

10.8.1.1 Setting Up Your Electronic Signature

If you have not done so already, you need to set up your electronic signature. You do this from the **Info > Edit User Info** option at the top of the **PACT Menu Bar**.

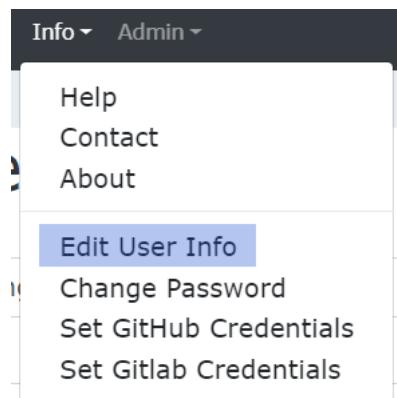


Figure 10-12. *Edit User Info* Menu Item

This brings up the **Editing User Page**.

The screenshot shows the 'Editing User' page in the PACT application. At the top, there's a navigation bar with links for Home, Log Out, Info, and Admin. Below it, a header says 'All Projects' and 'Editing User'. The form contains fields for Email (michelle@altech-marketing.com), First Name (Michelle), Last Name (Lange), Title (Contractor), Phone, Password, Time Zone (selected as '(GMT-05:00) Eastern Time (US & Canada)'), Signature File (Choose File: ML-Signature.jpg), Profile Picture (Choose File: No file chosen), and a checked checkbox for 'Notify on changes'. There's also an 'Organization' field containing 'patmos engineering services'. Under 'Role', 'AirWorthinessCert Member' is selected from a dropdown menu. Below the roles, there are three status options: 'User disabled' (disabled), 'Password reset required' (disabled), and 'FullAdmin' (selected). At the bottom are buttons for 'Update User' (highlighted in blue), 'Show', 'Change Password', and 'Back'.

Figure 10-13. Editing User Page

Note that this is very similar to the New/Edit User pages used by the Administrator to set up your PACT access. You can edit certain aspects of your information, such as your signature. PACT supports most graphic formats such as PNG, TIF, GIF or JPEG for your signature file. You can create a signature file by scanning your signature or taking a picture of your signature with a phone.

To add a signature file to your user record in PACT, scroll down to the **Signature File** field, click **Choose File**, and then navigate to and select your electronic signature file.

Next click the **Update User** button to save the signature to your user profile. This signature will then be used when you sign into a review.

10.8.1.2 Filling Out the Sign-In Sheet

When appropriate in your process (i.e., when you start the review, when you complete the review, or whenever your plans call for this), at the bottom of the **Editing Review Page** (see Figure 10-11), click the **Sign-In Sheet** button.

This brings up the **Sign-In Sheet**. Here you click **Sign Review** button to sign the sheet.

Attendee Sign-In Sheet

Title: Review of PHAC draft

Details:

Review ID: APC_CB-REVIEW-1
Project: ACMEY Company Avionics Passenger Counter

Description: Formal review of the Plan for Hardware Aspects of Certification (PHAC) 800-PHAC-01 rev 0.1.
Exit criteria for the review is to create the initial baseline of the PHAC at Revision A.

Review created by: Dave Newton
Evaluators who have signed: Michelle Lange
Dave Newton

Evaluators who have not signed:

Evaluation Date: 2020-06-25

Attendee List

Dave Newton, Senior FPGA Engineer

Michelle Lange, Contractor

Save Sign-In Sheet **Back**

Figure 10-14. Sign-In Sheet

Clicking **Save Sign-In Sheet** will download a PDF copy of the sign-in sheet.

10.8.2 Filling in the Checklist

After signing in, follow these steps to fill in the **Checklist** as part of the review:

- 1 Open Checklist:** In the Checklist portion of the **Editing Review Page**, click on the **Fill in Checklist** button. This opens your personal version of the Checklist.

The screenshot shows a web-based application titled "PACT" with a dark header bar containing the logo, "Home", "Log Out", "Info", and "Admin". Below the header, a breadcrumb navigation bar shows "All Projects / Flight Management Computer / More Hardware I/O / PHAC Review". The main content area is titled "Fill In Checklist" and features a search bar. A table lists four checklist items:

ID	Description	Applicable DAL	Reference	Supplements	Status	Notes
1	Does the PHAC include a System Overview (including a system functional description, system failure conditions, system architecture, a description of the allocation of the functions to hardware items and software) and references to existing system documentation?	A	10.1.1-1		Pass ▾	<input type="text" value="01_PHAC.POSTED.r4.doc"/> Document Comment Action Item
2	Does the PHAC include a Hardware Overview (including hardware functions, hardware items, architecture, new technologies to be used, and any fail-safe, fault tolerant, redundancy and partitioning techniques to be used)?	A	10.1.1-2		Pass ▾	<input type="text" value="01_PHAC.POSTED.r4.doc"/> Document Comment Action Item
3	For each digital or mixed-signal custom device, does the PHAC identify the development assurance level, the simple or complex classification, and if a device is classified as simple, the justification based on the simple classification criteria? For each simple device, does the PHAC or other appropriate plan include a definition of the device functions, a plan for	A	AC20-152A CD-1		Fail ▾	Document needs to include the mixed-signal device <input type="text" value="01_PHAC.POSTED.r4.doc"/> Document Comment Action Item

At the bottom of the page are three buttons: "Submit Changes", "Export", and "Back". The footer contains links to "Process and Artifact Compliance Tool", "About", and "Contact".

Figure 10-15. Filling in a Checklist

- 2 Read Question:** You first read the question on the left. The question presented applies to the design assurance level (DAL) of the Hardware or Software Item (as specified when it was initially created).
- 3 Refer to Reference if Needed:** If you do not understand the question, you can review the reference shown from the appropriate document (i.e., DO-254 for Hardware Item references or DO-178C for Software Item References).
- 4 Review Pertinent Supplements:** If the Software Item is subject to DO-178C compliance and you are using Supplements as part of this process, the supplements that are applicable to the question are listed. DO-254 (Hardware Items) do not have supplements.
- 5 Provide Status for Question:** After you have had a chance to evaluate if the document/artifact under review meets the question, you set a Status: pass, fail or NA. If a question fails or is not applicable, you must specify some Notes as to why.
- 6 Repeat for Each Question:** Continue this process to apply a status to each checklist question.
- 7 Track Issues:** Each question that fails should have an appropriate Document Comment or Action Item to track it to resolution. (See Section 10.9 for more information).

- 8 Submit Your Checklist Input:** When you are done, click **Submit Changes** button. This captures each status you selected and puts you into a status viewing page for the review.
- 9 See Combined Evaluator Input:** To see the entirety of all Evaluators review status, select **Consolidated Checklist**. This brings you to a view of the checklist where you can see every Evaluator's status and notes.

10.9 Tracking and Resolving Issues from a Review

Questions that fail in the checklist are review issues that require tracking and resolution. You can track an issue by one of two means: Action Items or Document Comments.

10.9.1 Action Items

If you want to capture an Action Item (AI), which is more than just a comment about a document, you do so by clicking the Action Item button, to the right of the checklist status field. This brings you to the **Action Items List Page**.

The screenshot shows the PACT application interface for managing action items. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, Action Items, Undo, and Back. Below the navigation is a breadcrumb trail: All Projects / Flight Management Computer / A Hardware Module / PHAC Review. The main title is "Action Items List". On the right, there is a search bar labeled "Search:". The table below lists action items with columns for Action Item ID, Description, Opened By, Assigned To, Status, Note, and three buttons (Show, Edit, Delete). One row is shown, detailing an action item for a non-compliance issue on the PHAC review, assigned to Dave Newton and Michelle Lange, with a status of Open and a note indicating changes were made. At the bottom of the table are buttons for New Action Item, Back, and Undo.

Action Item ID	Description	Opened By	Assigned To	Status	Note	Show	Edit	Delete
1	You must take action regarding non-compliance issue #2 on PHAC review.	Dave Newton	Michelle Lange	Open	ML: 4/20/20 I made the appropriate changes. See page 67 of PHAC.			

Figure 10-16. Action Items List Page

From here you click **New Action Item** to bring up the **New Action Item Page**.

The screenshot shows the 'New Action Item' page in the PACT system. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Action Item. Below that is a breadcrumb trail: All Projects / Flight Management Computer / Action Items. The main title is 'New Action Item'. There are two large text areas with rich text editors, one for 'Description' and one for 'Notes'. Each editor has a toolbar with various icons for bold, italic, underline, etc. Below each editor is a dropdown menu. Other fields include 'Action Item ID' (set to 2), 'Opened By' (set to Michelle Lange), 'Assigned To' (set to Michelle Lange), and 'Status' (set to Open). At the bottom are 'Save Action Item' and 'Back' buttons.

Figure 10-17. New Action Item Page

This is similar to a scaled down version of the Problem Reporting system. It is simplified because its focus is on capturing issues that need to be addressed within the scope of a Review. The pertinent fields you must fill in are as follows:

- **Action Item ID:** This is assigned by the tool and it is the next available number.
- **Description:** Enter a description of what the Action is.
- **Opened By:** This field defaults to the person who is signed in but you can change it if you are opening an Action Item for someone else.
- **Assigned To:** This field defaults to the person who is signed in but you should change it to the person who should perform the needed action.
- **Status:** These are the same as for the PR system.
- **Notes:** Here the person who addresses the action can capture notes about corrective actions taken, verification activities related to the action, etc.

When a person is Assigned an AI, they will get an email notification similar to the following:

Dear Michelle Lange:

A new Action Item has been assigned to you. Please see below for the details.

Detail:

Project: Flight Management Computer

Item/Artifact: I/O Board

Review: PSAC Review

Description: Get buy off from team on Cert action plan

Opened By: Michelle Lange

Status: Open

Notes:

Action Item Link: https://acs-pact.com/pact_awc/reviews/216/action_items/58

PACT Automation

pact_cm@airworthinesscert.com

Figure 10-18. Sample Email Notification for Action Items

The Assigned person would then perform the action required. When finished, they would navigate to the Item, Edit the specific Review, fill in the Checklist, and click on the Action Item to Edit it.



Tip: If you do not receive such an email, check to see if “Notify on Changes” is turned off in your User Profile. Alternately, the email may have been diverted to your Junk Mail account. Its best practice to set up pact_cm@airworthinesscert.com as a “safe sender.”

The screenshot shows the 'Editing Action Item' page in the PACT system. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Action Item. Below that is a breadcrumb trail: All Projects / Flight Management Computer / Action Items. The main area is titled 'Editing Action Item'. It contains several input fields and rich text editors:

- * Action Item ID: A text input field containing '1'.
- * Description: A rich text editor toolbar followed by a text area containing 'Get with Cert Authority to hash out Cert Liaison plan.'
- Opened By: A dropdown menu showing 'Michelle Lange'.
- Assigned To: A dropdown menu showing 'Michelle Lange'.
- Status: A dropdown menu showing 'Open'.
- Notes: A rich text editor toolbar followed by a text area.
- Review: A dropdown menu showing 'PSAC Review'.

At the bottom, there are buttons for Save Action Item, Show, Back, Undo, and a prominent blue 'Update Action item' button. Below the form, there are links for Process and Artifact Compliance Tool, About, and Contact.

Figure 10-19. Editing Action Item Page

The person who made the fix should make appropriate Edits to the Action Item indicating what was fixed and click **Update Action item**.



Note that the complete Action Item resolution process is very similar to the Problem Reporting system. However, since the scope is just addressing Review Issues (as opposed to capturing issues with the action Hardware/Software Item), you can perform a simplified process.

How you intend to use this Action Item system in the context of your Reviews should be documented in the appropriate plan (e.g., *Hardware Verification and Validation Plan*) for the Hardware/Software Item you are developing.

10.9.2 Document Comments

If a review issue is simply something that needs to change in a document, during the process of filling in a Checklist (see Section 0), you can click on **Document Comment** under the **Notes** area in the far right of the page.

This opens a new **Document Comments List Page**.

Document Comments List

Search:

Comment ID	Requested By	Assigned To	Status	Document Revision	Comment	Show	Edit	Delete
1	Steve Gregor	Steve Gregor	Open		this is a comment towards draft version 0.1. - here's a comment towards review #2.	Show	Edit	Delete
2	Steve Gregor	Steve Gregor	Open	0.1	comment here.	Show	Edit	Delete
3	Steve Gregor	Steve Gregor	Open			Show	Edit	Delete
4	Michelle Lange	Michelle Lange	Open	b	A problem on page 2	Show	Edit	Delete
5	Michelle Lange	Michelle Lange	Open	b	Missing certification liaison process. Should be added to section 7.	Show	Edit	Delete

[New Document Comment](#)

[Go to Document](#)

Process and Artifact Compliance Tool

[About](#) [Contact](#)

Figure 10-20. Document Comments List Page

Here you click the [New Document Comment](#) button which puts you in a **New Document Comment Page**.

The screenshot displays the 'New Document Comment' interface. At the top, a navigation bar includes links for Home, Log Out, Info, Admin, Document Comment, and Back. The main content area has a title 'New Document Comment' and a sub-title 'Document: 15_VV_HW_Evidence.POSTED.r1.xls'. Below this, several input fields are present: 'Document Comment ID' (value 1), 'Opened By' (selected value 'Paul Carrick'), 'Assigned To' (selected value 'Paul Carrick'), 'Status' (selected value 'Open'), and a 'Change Made In Document Revision' field which is currently empty. A 'Draft Version' field is also shown. A rich text editor toolbar is located above a large text area for the comment. At the bottom, a dropdown menu shows the document path: All Projects / Flight Management Computer / A Hardware Module / Documents / 15_VV_HW_Evidence.POSTED.r1.xls / Document Comments. Finally, there are 'Create Document Comment' and 'Back' buttons.

Figure 10-21. New Document Comment Page

Fill in the **New Document Comment Page** with as much information as possible to capture the issue and assign to the Team Member who is responsible for the change.

 Note that Document Comments are handled similarly to Action Items. While they are not as formally tracked and monitored, they still go through a resolution process that includes tracking the status to closure. How you intend to use this feature should be documented in your appropriate project plans.

10.10 Passing a Review

When all issues are resolved and verified by another engineer (usually not the one who implemented the corrective action), then the Review can resume and all items that previously

failed can be marked with a Pass status. This must be done by each Evaluator who initially marked a Fail, and the Consolidated Checklist will then display all Evaluators passing every checklist item. When this occurs, the document under review is now a *Release Candidate*. See Section 5.2 to understand what a Release is in terms of PACT versioning.

10.11 Exporting Checklists

During or after a review, you may need to Export one or more Checklists. You access this function when you are filling in the Checklist itself from within the **Edit Review Page**.

Fill In Checklist

ID	Description	Applicable DAL	Reference	Supplements	Status	Notes
	b. System Architecture c. Processors used d. HW/SW Interfaces e. Safety Features			Formal Method		Document Comment Action Item
2	Does the PSAC contain a Software Overview that describes the Software functions with an emphasis on the proposed safety and partitioning concepts (Ex. Resource sharing, redundancy, fault tolerance, mitigation of single event upsets, and timing and scheduling strategies.	B	11.1.b	Model Based Object Oriented Formal Method	Pass	S_01_C_PSAC-DOWNLOAD.r2.doc Document Comment Action Item
3	Does the PSAC describe the Certification Considerations including the following: a. Certification Basis and Means of Compliance b. Proposed software levels of each function c. Summary of the justification for those levels per the preliminary SSA d. Any applicable IPs/CRIs or other agreements	B	11.1.c	Model Based Object Oriented Formal Method	Pass	S_01_C_PSAC-DOWNLOAD.r2.doc Document Comment Action Item
4	Does the PSAC describe the Software Life cycle including the following: a. Summary of each life cycle process that explains how the DO-178C objectives of each life cycle process	B	11.1.d	Model Based Object Oriented Formal Method	Fail	Missing certification liaison process S_01_C_PSAC-DOWNLOAD.r2.doc

Search:

[Submit Changes](#) [Export](#) [Back](#)

Figure 10-22. Exporting a Checklist

Click the **Export** button. This brings up a new **Export Checklist Page** in a new browser tab.

As with all other Export capabilities, you select the format and then click **Export**. Do not forget to close the new tab.

11 Managing Problem Reports

Verifying that a Project and its corresponding Hardware/Software Items are specified correctly, verified fully, and performing their intended function (with no unintended functions) is vital to any safety-critical process. Likewise, it is essential that the planned and approved processes be followed, and the corresponding documentation be correct and complete. When any deviations of requirements, development or verification process, or documentation are found, the team member who discovers the issue must file a Problem Report.

PACT supports a Problem Report feature that helps team members capture, process, track (to resolution), and report on problems. These features and how to use them are described in the following subsections. The first subsection talks about the Problem Reporting Process itself, and the remaining subsections describe how PACT supports this.

11.1 Understanding the Problem Reporting Flow

Figure 11-1 graphically depicts a typical PR process flow within a project.

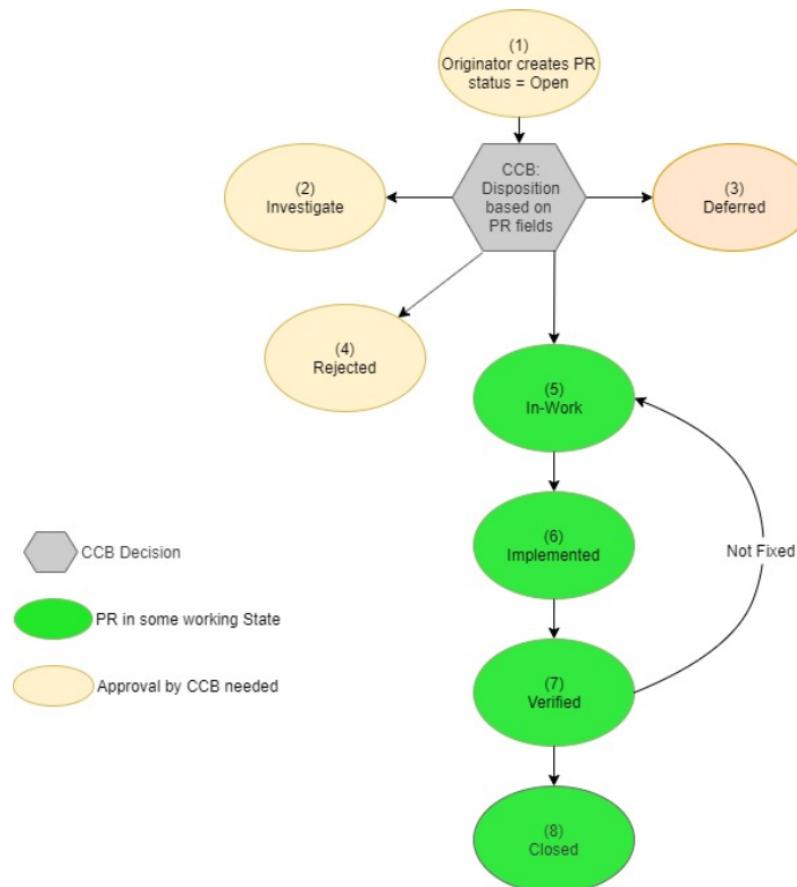


Figure 11-1. Problem Report Flow

The flow involves the following steps (which are indicated by the number in parentheses in the descriptions):

- 1 Open:** The first step in the Problem Reporting flow is to Open a PR. Problem reports are initiated from many sources. Some come from the product and process development teams. Some come from internal or external verification teams. Some come from customers. When a PR is opened, it is assigned a unique tracking ID and the Open status. An Open PR should be assigned to a Change Control Board (CCB) representative. Many safety-critical programs create a group of subject matter experts (i.e., the CCB), who make decisions regarding whether or not proposed changes should be made to the project, item or product. The CCB will examine the PR and disposition it to determine what happens next. See Section 0 for how to Open a New PR. Each of the following steps are accomplished by Editing a PR. See Section 11.4 for how to Edit an existing PR.
- 2 Investigate:** The CCB may determine that more investigation is needed to determine the appropriate course of action for the PR. This would set the status to Investigate and assign the appropriate person to do this investigation.
- 3 Deferred:** The CCB (and/or person who was assigned to Investigate) may decide that the PR should be deferred to a later time.
- 4 Rejected:** The CCB (and/or person who was assigned to Investigate) may decide that the PR is not a valid problem and should be rejected.
- 5 In-Work:** The CCB (and/or person who was assigned to Investigate) may assign the Problem to a specific department (typically engineering) and individual for resolution. The CCB also assigns a Target Completion Date.
- 6 Implemented:** Once the engineer assigned to work on the PR feels they have resolved it, they edit the PR to document the corrective actions they took and the targeted release for the fix.
- 7 Verified:** Following the implementation activities, an independent person or group is assigned to verify the implementation. The results of the verification activity are recorded in the Problem Report. If the implementation to fix the PR is found to be inadequate, the verifier sets the PR status back to “In-Work” and re-assigns the implementing engineer.
- 8 Closed:** Once the PR is implemented and verified, it is closed. Some processes may require that PRs only be Closed with CCB approval.
 - For a PR that affects a production released product configuration, the PR is listed on Engineering Change Notice (Release) Form and the PR should list the target Change Order.
 - PRs that do not affect a production released product are closed within the PR record itself.



Note the process shown here is recommended. However, the PACT Problem Reporting feature is flexible enough to allow a variation of processes for handling Problem Reports.

11.2 Listing Problem Reports

The **Project Level Problem Reports List** is accessed through the **Navigation** column of a particular Project (or through the **Navigation** section of a specific **Project Page**). This lists all Problem Reports for an entire project.

The **Item Level Problem Reports List** is accessed through the **Navigation** column of a particular Hardware/Software Item (or through the **Navigation** section of a specific **Hardware/Software Item Page**). This lists the Problem Reports for a specific Hardware/Software Item. Regardless of how you access problem reports the interface is identical.

When you choose to view **Item Level Problem Reports** you will be taken to the **Problem Reports List Page** showing the Problem Reports associated with the Item (skip to Section 11.2.2). When you choose to view **Project Level Problem Reports** you will be taken to the **Select Problem Reports Page** as shown here.

All Projects / Process and Artifact Compliance Tool / Problem Reports

Select Problem Reports

Field:

Value:

Show Problem Reports New Problem Report PRs Opened by me PRs Assigned to Me Back

Figure 11-2. Select Problem Reports Page

On the **Select Problem Reports Page** you can do the following:

- Show Problem Reports
- Create a new Problem Report (See section 11.2)
- View PRs that you have opened
- View PRs assigned to you.

11.2.1 Choosing Problem Reports to View

PACT provides a lot of flexibility in terms of filtering PRs. This is useful when a program has a lot of PRs and you want to look at only a certain subset of them for some particular purpose. There are several ways to filter the list of Problem Reports.

At the top of the **Select Problem Reports Page**, you have several Filtering options as follows:

- **Field:**

The **Field** menu lets you choose a field to search. The fields correspond to the information provided in each field when the PR was created.

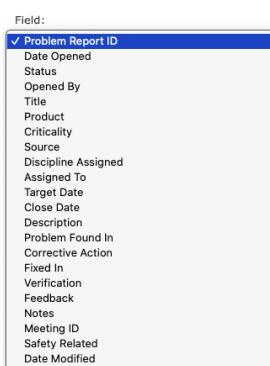


Figure 11-3. Field Drop Down List Example

- **Value:** Permits you to specify the value on which to search. PACT searches the **Field** for any occurrence of the **Value**. For example, you can select **Problem Report ID** as the type of **Field** and then enter in 12 in the **Value** field. Then any Problem Report with an ID containing 12 will be listed when you click **Show Problem Reports**. (Note that if you had PRs with ID 12, 112, 128, or 1112, all of those would be listed).

To select all problem reports simple click **Show Problem Reports** without entering anything in **Value**.

You can select PRs in other ways as well:

- **View all PRs You Have Created:** To view all Problem Reports you have created, click the **My Problem Reports** button.
- **View All PRs Assigned to You:** To view all Problem Reports currently assigned to you, click the **PRs Assigned to Me** button.

11.2.2 Problem Reports List

No matter how you choose the Problem Reports to view, they will appear in the **Problem Reports List Page**.

Problem Report ID	Title	Description	Hardware/Software Item	Opened By	Assigned To	Status	Safety	Criticality	Datemod	History	Show	Edit
1	Issue with software signal IN_2	Software signal IN_2 is mismatched. Should be IN_02 as per requirements.	I/O_Software	Michelle Lange	Steve Gregor	Open	<input type="checkbox"/>	Type 1A - Failure (No Safety Impact)		History		
2	REQ-002 not clear	REQ-002 needs to be re-evaluated and clarified so it is understood and verifiable.	NAV_fpga	Michelle Lange	Steve Gregor	Closed	<input type="checkbox"/>	Type 2 - Other Fault (No Failure)		History		
4	A new problem	Check out the problem.	I/O_Software	Michelle Lange	Tammy Reeve	Open	<input type="checkbox"/>	Type 2 - Other Fault (No Failure)		History		
5	Correct the requirement I/O_Software-HLR-0009 to use the updated SSM value.	Update the high level requirement I/O_Software-HLR-0009 to use the newly specified SSM of 11b instead of 10b.	I/O_Software	Steve Gregor	Michelle Lange	Implemented	<input type="checkbox"/>	Type 4 - Enhancements or Documentation		History		
6	CAN bus flush needed prior to normal CAN rx operation.	A new low level requirement needs to be added to address manufacturer errata on the CAN bus peripheral. The RX buffer needs to be read prior to initial launch of the receive task during initialization.	I/O_Software	Steve Gregor	Michelle Lange	Open	<input type="checkbox"/>	Type 4 - Enhancements or Documentation		History		
7	I2C read address update from H/W.	Update to schematic show new I2C receive address to I/O board. New address is now 0x25. All incoming I2C messages to I/O board shall now be accepted if the 0x25 address is present. Software must be updated to read new address.	I/O_Software	Steve Gregor	Michelle Lange	Open	<input type="checkbox"/>	Type 4 - Enhancements or Documentation		History		
8	Deleted		I/O_Software				<input type="checkbox"/>			History		
9	There is another problem	There is another problem.	AHM	Michelle Lange	Michelle Lange	Open	<input type="checkbox"/>	Type 1A - Failure (No Safety Impact)		History		
10	Found a problem with XYZ	This is a description.	CommBrd_fpga	Michelle Lange	Michelle Lange	Open	<input type="checkbox"/>			History		

New Problem Report | Export | Import | All Problem Reports | PRs Opened by me | PRs Assigned to Me | Back

Process and Artifact Compliance Tool | About | Contact

Figure 11-4. Problem Reports List Page

You can further explore any of these listed Problem Reports by clicking the **Show** icon. Likewise, you can **Edit** (see Section 11.4) or **Delete** (see Section **Error! Reference source not found.**) a Problem Report from these corresponding icons.

At the top of the **Problem Reports List Page** you have several Filtering options as follows:

- **Field/Value:** As described in the previous section.
- **Search:** Here you can type in a text string to help you find PRs dealing with a certain issue or aspect of the project. For example, you could type in “inputs” and hit return and all PRs that reference “inputs” will be listed. Similarly, you could type in a team member’s name, such as “John” and hit return, and the list of all PRs where John has been assigned or mentioned in any way will show.

11.3 Creating a New Problem Report

Clicking the **New Problem Report** button from the **Problem Reports List** brings up the **New Problem Report Page**. We are showing and describing it here in two parts due to the many fields in the page.

The screenshot shows the 'New Problem Report' page in the PACT application. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Problem Report. Below that, a breadcrumb trail shows 'All Projects / ACMEY Company Avionics Passenger Counter / Problem Reports'. The main title 'New Problem Report' is centered above a form. The form includes fields for 'Project' (set to 'ACMEY Company Avionics Passenger Counter'), 'Problem Report ID' (set to '5'), 'Title' (empty), 'Problem Report Applies to' (a dropdown menu), 'Date Opened' (set to '08/11/2020 03:53:35 PM'), 'Opened By' (set to 'Michelle Lange'), 'Assigned To' (set to 'Michelle Lange'), 'Status' (set to 'Open'), 'Date Modified' (set to '08/11/2020 03:53:35 PM'), 'Product' (empty), 'Criticality' (a dropdown menu), 'Source' (set to 'Internal'), 'Discipline Assigned' (set to 'Engineering'), 'Target Date' (a date input field), and 'Close Date' (a date input field). Most fields have dropdown arrows indicating they can be expanded.

Figure 11-5. New Problem Report Page (Top Half)

To add a new Problem Report (PR), fill in the fields on this page as follows:

- **Project:** This is filled in to show the Project to which this PR applies.
- **Problem Report ID:** This is the next available number for the PRs filed against this Project or Item.
- **Title:** This is a freeform field where you enter a descriptive title for the problem report.

- **Problem Report Applies To:** In this field you choose the specific Hardware/Software Item for the Project to which the PR applies. Any Documents, Source Code or Test Procedures that this Problem Report applies to can be referenced in this section. When a Hardware/Software Item is selected, several other fields will appear. An Artifact Type list permits the selection of specific artifacts (Documents, Source Code or Test Procedures). Once selected, the artifacts will appear in the list below. The artifacts can then be selected. Clicking the Add button will add the selected artifacts into the Selected Artifacts list that are associated with this Problem Report. In the Selected Artifacts list, selecting the artifacts and clicking Remove will remove the artifacts from the list.

Problem Report Applies to

Process and Artifact Compliance Tool

Artifact Type

Documents

Add

Selected Artifacts

Remove

In addition to Documents, Source Code and Test Procedures in the Artifact Type list the type of Other can be selected which permits entering a text field to describe any other artifact associated with this PR.

Problem Report Applies to

Process and Artifact Compliance Tool

Artifact Type

Other

Other Artifacts

- **Date Opened:** This field sets to the date the PR was initially opened. You cannot override this.
- **Opened By:** This defaults to the current user, but you can override this if you are opening the PR on behalf of another team member.
- **Assigned To:** This defaults to the current user, but you can override this is to assign it to a different team member (or the CCB board representative if you are following the flow of Figure 11-1. Problem Report Flow). The assigned user will receive an email notifying them of the assigned PR.
- **Status:** The status is always Open for a new problem report. As the PR goes through various stages, the assigned user will change the status. The status options align with the PR life cycle shown in Figure 11-1.
- **Date Modified:** This field is greyed out for a New PR but is filled in later in the process when the PR transitions through its various states.
- **Product:** If this PR applies to a specific Product, you can enter that here.
- **Criticality:** Select the criticality from the criticality types listed as follows:
 - Type 0 – Failure (Safety Impact)
A problem whose consequences result in a failure condition with a safety impact, or even possible safety impact
 - Type 1A – Failure (No Safety Impact)
A problem whose consequences include a failure with NO adverse safety impact
Reliability is degraded.
 - Type 1B – Failure (Insignificant Consequences)
A non-significant functional failure with no significant functional consequence.
 - Type 2 – Other Fault (No Failure)
A functional defect that does not result in significant failure and does not fit well into any other category.

- Type 3A -- Process (Significant Deviation)
A significant deviation whose effects could be to lower the assurance that the software behaves as intended and has no unintended behavior.
- Type 3B – Process (Insignificant Deviation)
A non-significant deviation that does not affect the assurance obtained.
- Type 4 – Enhancements or Documentation
Any initial development, updates, or enhancements to the software or documentation.



Note that these “Types” are defined as per the aviation industry document, AC 20-189, Management of Open Problem Reports.

- **Source:** Select the source who discovered the PR. You can choose from either Internal, Customer, Cert Authority or Other.
- **Discipline Assigned:** Choose from Engineering, Manufacturing, Quality or Other.
- **Target Date:** This is the target date by which the problem should be resolved. It is usually filled in by the CCB during their evaluation of the PR.
- **Close Date:** This is greyed out when the PR is opened and filled in by the tool when the PR enters the **Closed** status.

The screenshot shows the bottom half of a 'New Problem Report' form. The sections include:

- Description:** A large text area with a toolbar for bold, italic, underline, font size, and other text styling.
- Found In:** A text input field.
- Fixed In:** A text input field.
- Corrective Action:** A large text area with a toolbar for bold, italic, underline, font size, and other text styling.
- Verification Comments:** A large text area with a toolbar for bold, italic, underline, font size, and other text styling.
- Meeting ID:** A text input field.
- Safety Related:** A checkbox labeled 'Safety Related'.

At the bottom are two buttons: 'Create Problem Report' and 'Back'.

Figure 11-6. New Problem Report Page (Bottom Half)

- **Description:** You should enter a description for a Problem Report when you open it. Be as precise as possible to aid with recreation and resolution. Once the PR is approved and moves to the “In-Work” state, this field becomes non-editable.
- **Found In:** Add a description as to where (e.g., a certain file, specific operation, document, etc.) you found the problem.

- **Fixed In:** Add a description of the version that the fix is incorporated.
- **Corrective Action:** This field is not editable for New PRs but should be filled in by the engineer that implements the fix to describe what was done.
- **Verification Comments:** Similar to the field above, this field is not editable for New PRs but should be filled in by the team member who verifies the fix to describe what was done and found.
- **Meeting ID:** You may also enter a meeting ID if the problem was discussed in a meeting.
- **Safety-Related:** While the Criticality provided some information about safety impact, click here to ensure that safety-critical PR get flagged as having a safety impact.



Note that numerous fields that are greyed out within this page. Do not concern yourself with these if you are creating a New PR. As described above, these fields are used only when you edit an existing PR. See Section 11.4 for more information.

Additionally, a file may be attached to this Problem Report. Clicking the [Add File](#) button will permit a file to be attached to the Problem Report.

Select [Create Problem report](#) to create the Problem Report or [Back](#) to exit without saving the Problem Report and bring you back to the **Problem Reports List Page**.

When you create a PR, the person assigned as well as the Configuration Manager will be notified by email. As the PR creator, you will be notified any time the PR changes. This is true also for the Configuration Manager. The notification will look similar to this:

Dear John Doe:

A problem report has been changed. Please see below for the details.

Detail:

Project: Flight Management Computer
Item/Artifact: I/O Board
ID: 5
Title: Correct the requirement I/O_Software-HLR-0009 to use the updated SSM value.
Description: Update the high-level requirement I/O_Software-HLR-0009 to use the newly specified SSM of 11b instead of 10b.
Opened By: Jane Doe
Status: Implemented
Problem Report Link: https://acs-pact.com/pact_awc/projects/21/problem_reports/247

PACT Automation

pact_cm@airworthinesscert.com

Figure 11-7. PR Notification Email Example

Tip: If you do not receive such an email, check to see if “Notify on Changes” is turned off in your User Profile. Alternately, the email may have been diverted to your Junk Mail account. It’s best practice to set up pact_cm@airworthinesscert.com as a “safe sender.”

11.4 Editing a Problem Report

If you are assigned a Problem Report, you will receive an email notification.

A new problem report has been assigned to you.

 pact_cm@airworthinesscert.com <pact_cm@airworthinesscert.com>

To: [Paul Carrick](#); Cc: [Paul Carrick](#)

Tuesday, June 30, 2020 at 11:44 AM

Dear Paul Carrick:

A new problem report has been created. Please see below for the details.

Detail:

Project:	Process and Artifact Compliance Tool
Item/Artifact:	Process and Artifact Compliance Tool
ID:	296
Title:	Unlinked Trace Items Needs Instruction

Paul, Tammy suggested I open a PR and NOT defer it. She agreed that the “Unlinked”button in the Requirements Traceability Matrix could use explanation to make it more intuitive. And the more I thought about it, its a general issue on that age. When you get to the RTM page, nothing is checked by default as follows:

All Projects / ACMEY Company Avionics Passenger Counter / _Avionics Passenger Counter (APC) Controller Board

Requirements Tracing

System Requirements Requirements Test Cases Test Procedures
 Conceptual Design Source Code

Forward Reverse

[View](#) [Unlinked](#) [Derived](#) [Export](#) [Back](#)

Process and Artifact Compliance Tool

We felt either ALL items should be checked on that page by default OR if you click, View, Export, Derived or Unlinked buttons with NOTHING selected, there should be a message that says “You must select the items for which you want the traceability report” (or something like this). Because right now if you don’t select anything, you simply get a blank Traceability page with nothing on it. That’s not intuitive or helpful.

Alternately, these instructions could be on the page that is drawn, but that involves the user having to X out of that browser window (there is no Close button at the bottom) and go back and do it. So ideally message/instructions should be on that original page.

Opened By: Michelle Lange
Date Opened: 2020-06-30 12:00AM EDT
Status: Open
Product: PACT
Criticality: Type 2 - Other Fault (No Failure)
Safety Related: false
Source: Internal
Problem Found In: Engineering
Discipline Assigned: Engineering

Figure 11-8. PR Notification Email Example

Likewise, if you opened a PR, you would receive similar emails letting you know when the PR has changed states.

Click the link in the email notification to see the PR. When you have taken the appropriate action, you then go and edit the PR.

To edit a Problem Report, from the **Problem Reports List**, click the  icon for a specific Problem Report (or choose the [Edit](#) button from a specific **Problem Report Page**).

The **Editing Problem Report Page** itself is nearly identical to the **New Problem Report Page**. The difference is only in which fields are active, since different information is required for New PRs versus PRs that are progressing through the resolution process. Each of the fields were already described.

When you are assigned a PR, you will usually have to go in and first read the PR to understand the situation and then take the appropriate action. Once you have taken that action, you will go in and Edit the PR to update it with the pertinent information about what was done. What follows are the typical fields that you will fill in while editing an existing Problem Report.

- **Assigned to:** Here you will assign the next person who needs to take some action on the PR.
- **Status:** Here you will change the status once you are done performing your assigned action.
- **Date Modified:** This field updates automatically to capture the date when the PR was modified.
- **Corrective Action:** If you have implemented a fix, enter information on the corrective actions taken to address the problem report.
- **Verification Comments:** If you have verified a fix, enter information on what you verified, how you did it and what you found.

11.5 Viewing a Problem Report



Clicking the Show icon in the Proport list will let you view the detail for a specific Problem Report.

The screenshot shows a software application window titled "Problem Report ID: APC-PR-1". The window has a dark header bar with the PACT logo and navigation links: Home, Log Out, Info, Admin, Problem Report, Edit, Email, and Back. Below the header, the URL is displayed as All Projects / ACMEY Company Avionics Passenger Counter / Problem Reports. The main content area is a blue-bordered box containing the following information:

Details:
Title: PHAC PR1
Referenced Artifacts:
Date Opened: 2020-06-24 09:00PM PDT
Opened By: John Public
Status: In-Work
Date Modified: 2021-02-23 09:32AM PST
Product: Avionic Passenger Counter
Criticality: Type 3A - Process (Significant Deviation)
Source: Internal
Discipline Assigned: Engineering
Assigned To: Michelle Lange
Target Date: 2020-07-05 09:00PM PDT
Close Date:
Description:
PHAC has a problem. On page 3.
Checking date change is correct.
Problem Found In: 800-PHAC-01 rev A
Corrective Action:
Fixed In:
Verification:
Meeting Id:
Safety Related: false
History: View
Files:

File	Uploaded

At the bottom of the window are four buttons: Edit, New Problem Report, Email Problem Report, and Back. The footer contains copyright information: Process and Artifact Compliance Tool (C) Airworthiness Certification Services, LLC PMB#224 5500 Olympic Drive Suite H-105 Gig Harbor Wa 98335 Phone/fax: 425-427-1956. It also includes links to About, Privacy and Security, and Contact.

11.6 Emailing Problem Reports

While viewing or editing a Problem Report you may email the problem report by clicking the **Email Problem Report** button. A new dialog will appear.

The screenshot shows a web-based application interface for PACT. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and Back. Below the navigation, the URL is displayed as All Projects / ACMEY Company Avionics Passenger Counter / Problem Reports. The main content area contains fields for 'Recipients' (with a placeholder '...'), 'Cc list' (another text input field), and a 'Comment' section which includes a rich text editor toolbar with various formatting options like bold, italic, underline, etc. At the bottom of the dialog are two buttons: 'Send Email' and 'Back'.

Process and Artifact Compliance Tool (C)
Airworthiness Certification Services, LLC
PMB#224
5500 Olympic Drive
Suite H-105
Gig Harbor Wa 98335
Phone/fax: 425-427-1956

About Privacy and Security Contact

Enter the email address(es) of the people you want to receive the problem report. The email addresses should be separated by commas. You can also enter and email addresses of people you want to be CCed in the CC List (also separated by commas). You may additionally enter a comment for the email. When everything has been entered click on **Send Email** to send the email.

11.7 Exporting Problem Reports

Exporting PRs gives you a way to view them outside of the PACT environment.

If you select **Export** from the **Problem Reports List Page**, PACT opens the **Export Problem Reports Page** in a new tab.

The screenshot shows a dark-themed web interface for the PACT Process and Artifact Compliance Tool. At the top, there's a navigation bar with links for Home, Log Out, Info, Admin, and a globe icon. Below the navigation is a breadcrumb trail: All Projects / Process and Artifact Compliance Tool / Problem Reports. The main title is "Export Problem Reports". A dropdown menu labeled "* Export type" is open, showing "HTML" as the selected option. Below the dropdown are two buttons: "Export" (in white text) and "Back" (in white text). The URL at the bottom of the page is "Process and Artifact Compliance Tool".

Figure 11-9. Export Problem Reports Page

As with all the other types of Export operations, depending on chosen format, the information will either display in this new page or a file will download to your computer. Close the **Export Problem Reports Page** to ensure you do not end up with multiple PACT session tabs.

11.8 Importing Problem Reports

Importing Problem Reports lets you to load PRs for a Hardware/Software Item from another application into the PACT environment. You can import PRs from a Comma Separated Value (CSV) file or Excel File (XLS or XLSX).

The file should only contain data (**no headers or extraneous entries**) with each line having the format of:

```
item_id,prid,dateopened,status,openedby,title,product,criticality,source,discipline_assigned,assignedto,target_date,close_date,description,problemfoundin,correctiveaction,fixed_in,verification,feedback,notes,meeting_id,safetyrelated,datemodified
```

To import Problem Reports, from the **Problem Reports List**, click the **Import** button. This brings up the **Import Problem Reports Page**.

* Project select

Process and Artifact Compliance Tool

Choose File No file chosen

Duplicates permitted

Load Problem Reports Back

Process and Artifact Compliance Tool

Figure 11-10. Import Problem Reports Page

From here, you choose the file including the Problem Reports you would like to import.

Click **Duplicates Permitted** checkbox if you want to permit duplicate IDs to be imported. If you do not check **Duplicates Permitted** and a Test Procedure already exists with the same ID as in the file when the file is imported, it will cause an error and the file will not be imported.

11.9 Viewing Problem Report History

Every time a Problem Report changes, these changes are recorded to capture how the Problem Report has changed over time.

To view the history for a specific Problem Report, from the **Problem Reports List**, click **History** in the **History** column for the Problem Report for which you want the history. This will produce a **Problem Report History List Page** as follows.

Action	Modified By	Status	Severity Type	Datemod	
	steve@patmos-eng.com	Open		2020-01-14 02:06PM EST	Show
	paul@patmos-eng.com	Open		2020-01-14 02:06PM EST	Show
	paul@patmos-eng.com	Assigned		2020-01-14 02:06PM EST	Show
	paul@patmos-eng.com	Implemented		2020-01-14 02:06PM EST	Show
	steve@patmos-eng.com	Implemented		2020-01-14 02:06PM EST	Show
	paul@patmos-eng.com	Verified		2020-01-14 02:06PM EST	Show

New Problem Report History Back to Problem Report Undo

Figure 11-11. Problem Report History List

Each line represents some sort of change to the PR, such as a new person assigned, a Status change, comments added, and so on.



To see more information at any stage of the history, click the [Show](#) icon in the appropriate row. From the **Problem Report History List**, you may go back to viewing the specific Problem Report for which you are viewing history, or you may go back to the list of all PRs.

12 Producing Certification Paperwork

At times you may be required to create a package of documents for your Hardware or Software Item. Whether it is providing detailed mid-program status for your customer, or the final paperwork required by the certification authority, PACT provides a simple way to select and package the documents that provide the crucial details of your project.

To create a certification documentation package, navigate to and select **Documents** under the specific **Hardware/Software Item**.

This brings up the **Documents List Page**.

Document ID	Name	Control Category	Revision	Revision Date	(Open/Closed) Comments	Show	Delete	Upload	Download
S_01_C_PSAC.DOWNLOAD.r2.doc	S_01_C_PSAC.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_02_C_SDOP.DOWNLOAD.r2.doc	S_02_C_SDOP.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_03_C_SVR.DOWNLOAD.r3.doc	S_03_C_SVR.DOWNLOAD.r3.doc				(0/0) Comments	Show	Delete	Upload	Download
S_04_C_SCMP.DOWNLOAD.r2.doc	S_04_C_SCMP.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_05_C_SQAR.DOWNLOAD.r2.doc	S_05_C_SQAR.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_06_C_SRS.DOWNLOAD.r2.doc	S_06_C_SRS.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_07_C_SDOS.DOWNLOAD.r2.doc	S_07_C_SDOS.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_08_C_SCS.DOWNLOAD.r2.doc	S_08_C_SCS.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_09_C_SRD.DOWNLOAD.r2.doc	S_09_C_SRD.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_10_C_SDD.DOWNLOAD.r2.doc	S_10_C_SDD.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_13_SVCP.DOWNLOAD.r1.doc	S_13_SVCP.DOWNLOAD.r1.doc				(0/0) Comments	Show	Delete	Upload	Download
S_14_C_SCAR.DOWNLOAD.r2.doc	S_14_C_SCAR.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_15_C_SECI.DOWNLOAD.r2.doc	S_15_C_SECI.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_16_C_SCI.DOWNLOAD.r2.doc	S_16_C_SCI.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_20_C_SAS.DOWNLOAD.r2.doc	S_20_C_SAS.DOWNLOAD.r2.doc				(0/0) Comments	Show	Delete	Upload	Download
S_21_CIA_DOWNLOAD.r1.doc	S_21_CIA_DOWNLOAD.r1.doc				(0/0) Comments	Show	Delete	Upload	Download
S_22_VV_SW_Evidence.DOWNLOAD.r1.xls	S_22_VV_SW_Evidence.DOWNLOAD.r1.xls				(0/0) Comments	Show	Delete	Upload	Download
S_23_SCP.DOWNLOAD.r1.doc	S_23_SCP.DOWNLOAD.r1.doc				(0/0) Comments	Show	Delete	Upload	Download

New Document Package Certification Documents Back Undo

Process and Artifact Compliance Tool About Contact

Figure 12-1. Documents List Page

At the bottom of this page, click on the **Package Certification Documents** button. This brings up the **Select Documents for Certification Package Page**.

Document ID	Name	Control Category	Revision	Revision Date	(Open/Closed) Comments
<input checked="" type="checkbox"/> S_01_C_PSAC.DOWNLOAD.r2.doc	S_01_C_PSAC.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_02_C_SDG.DOWNLOAD.r2.doc	S_02_C_SDG.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_03_C_SVP.DOWNLOAD.r3.doc	S_03_C_SVP.DOWNLOAD.r3.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_04_C_SCMP.DOWNLOAD.r2.doc	S_04_C_SCMP.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_05_C_SQAP.DOWNLOAD.r2.doc	S_05_C_SQAP.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_06_C_SRS.DOWNLOAD.r2.doc	S_06_C_SRS.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_07_C_SDS.DOWNLOAD.r2.doc	S_07_C_SDS.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_08_C_SCS.DOWNLOAD.r2.doc	S_08_C_SCS.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_09_C_SRQ.DOWNLOAD.r2.doc	S_09_C_SRQ.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_10_C_SDD.DOWNLOAD.r2.doc	S_10_C_SDD.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_13_SVCP.DOWNLOAD.r1.doc	S_13_SVCP.DOWNLOAD.r1.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_14_C_SCAR.DOWNLOAD.r2.doc	S_14_C_SCAR.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_15_C_SECI.DOWNLOAD.r2.doc	S_15_C_SECI.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_16_C_SCI.DOWNLOAD.r2.doc	S_16_C_SCI.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_20_C_SAS.DOWNLOAD.r2.doc	S_20_C_SAS.DOWNLOAD.r2.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_21_CIA.DOWNLOAD.r1.doc	S_21_CIA.DOWNLOAD.r1.doc				(0/0) Comments
<input checked="" type="checkbox"/> S_22_WW_SW_Evidence.DOWNLOAD.r1.xls	S_22_WW_SW_Evidence.DOWNLOAD.r1.xls				(0/0) Comments
<input checked="" type="checkbox"/> S_23_SCF.DOWNLOAD.r1.doc	S_23_SCF.DOWNLOAD.r1.doc				(0/0) Comments

Filename: Processor Board-documents.zip

[Create Certification Package](#) [Back](#)

[About](#) [Contact](#)**Figure 12-2. Select Documents for Certification Package**

If you want to package up the entire document set, select the checkbox at the top left of the page next to the **Document ID** heading. Otherwise, to select specific documents, click the checkboxes next to the documents you want included in the package.

In the **Filename** field, you can specify a name for the certification document page. If you do not specify a filename, PACT uses <Hardware/Software Item name>-documents.zip.

Then click the [Create Certification Package](#) button at the bottom of the page.

PACT produces the certification package zip file with the specified name in your computer's "Downloads" area.

13 Archiving Your Project Data

Projects may need to be archived to meet airworthiness or other safety-critical standards requirements. This section describes the various ways this can be achieved using PACT.

13.1 Understanding PACT Archiving

Archiving can serve multiple purposes and can occur at various levels.

You can take a snapshot of all your documentation and artifacts to capture the full status and description of your program at any time. This would involve exporting a copy of all the human readable information about your project, which could be used outside the tool. See Section 13.2 for more on this.

You can archive your project so that it can be resurrected within the PACT environment for potential future use. See Section 13.3 for more on this.

You can make your Project, along with the entire tool environment, available in your own local environment to have full control over the archive and/or resurrection of that development environment for potential future use. See Section 13.3 for more on this.

13.2 Exporting Documentation & Artifacts

PACT lets you Export any type of object stored in PACT. This ability lets you extract the information you need to cut and paste into the appropriate Document in your documentation set. For example, you can export all or some subset of your High-Level Requirements to a PDF format that you can paste into your *Software Requirements Document*. This document would be stored in PACT and controlled appropriately for your Project's needs.

However, you may also want to simply Export all the documentation and artifacts so it can be reviewed or stored outside of PACT for some other purpose.

You can create a full or partial download of all your Item's Documentation by creating a "Certification Package". See Section 12 for more information.

PACT also lets you export all of the following artifacts:

- System Requirements (See Section 4.2.8)
- Hardware/Software Item Requirements (See Section 4.3.5)
- Source Code (See Section 6.6)
- Test Cases (See Section 7.3.7)
- Test Procedures (See Section 7.4.7)
- Model Files (See Section 8.5)
- Requirements Traceability Matrices (See Section 9.3)

- Reviews/Checklists (See Section 10.11)
- Problem Reports (See Section 11.5)

13.3 Archiving Project Data & Tool

At the Project level, you can take a snapshot at any time or a full archive at the end of the Project by using the Project **Archives** function.

The **Archives** function is available from **Project List Page**.

The screenshot shows the PACT Project List Page. At the top, there is a navigation bar with links for Home, Log Out, Info, Admin, and Projects. Below the navigation bar, a search bar contains the text "All Projects". The main area is titled "Projects List". It features a table with columns for Identifier, Name, and Navigation. The Identifier column lists DSC, EFC, FMC, and PACT. The Name column lists Door and Slide Controller, Flight Controller, Flight Management Computer, and Process and Artifact Compliance Tool respectively. The Navigation column for each row provides links to "Hardware/Software Items", "System Requirements", "Problem Reports", "Review Status", and "Archives". The "Archives" link for the PACT project is highlighted with a red box and a mouse cursor icon pointing to it. At the bottom of the page, there are buttons for "New Project", "Back", and "Undo", and links for "About" and "Contact".

Identifier	Name	Navigation
DSC	Door and Slide Controller	Hardware/Software Items System Requirements Problem Reports Review Status Archives
EFC	Flight Controller	Hardware/Software Items System Requirements Problem Reports Review Status Archives
FMC	Flight Management Computer	Hardware/Software Items System Requirements Problem Reports Review Status Archives
PACT	Process and Artifact Compliance Tool	Hardware/Software Items System Requirements Problem Reports Review Status

Figure 13-1. Archives Function at the Project Level

When you click the **Archives** link on this page, this brings up the **Archives List Page**. This page shows all of the snapshots taken of the Project.

The screenshot shows the 'Archives List' page of the PACT system. At the top, there is a navigation bar with links for Home, Log Out, Info, and Admin. Below the navigation bar, a breadcrumb trail shows 'All Projects / Flight Management Computer'. The main content area is titled 'Archives List' and contains a table with the following data:

Name	Identifier	Description	Revision	Draft Version	Archived At						
FMC Project Archive	FMC_Project_Archive-01	FMC Project Archive After SDP Review		0.2	2020-07-17 08:37AM PDT	View Contents	Show	Edit	Delete		

Below the table, there are four buttons: 'New Archive' (highlighted in blue), 'Show Archives', 'Back', and 'Undo'. At the bottom of the page, there are links for 'About' and 'Contact'.

Figure 13-2. Archives List Page

Initially for a Project, this list will be blank.

On the **Archives List Page**, you can do the following:

- View Contents of an Existing Archive
- Edit the information for an existing Archive
- Delete an Archive
- Create a New Archive

To create a New Archive, click the **New Archive** button. This brings up the **New Archive Page**.

The screenshot shows the 'New Archive' page of the PACT application. At the top, there's a navigation bar with the PACT logo and links for Home, Log Out, Info, Admin, All Projects, Flight Management Computer, and Archives. The main title 'New Archive' is displayed prominently. Below the title are several input fields:

- * Name: FMC Project Archive
- * Archive ID: FMC_Project_Archive-01
- Description: FMC Project Archive After SDP Review
- Pact version: 1.7
- Revision: (empty)
- Version: 0.2
- Archived at: 07/17/2020, 08:37:23 AM

At the bottom of the form are two buttons: 'Create Archive' (highlighted in blue) and 'Back'.

Figure 13-3. New Archive Page

Fill out the Fields as follows:

- **Name:** Type in a descriptive name for the Archive.
- **Archive ID:** Create an ID that lets you identify the Archive.
- **Description:** Add any descriptive text that you want associated with the Archive.
- **Archived at:** This is the date of the Archive, captured by the tool.

Click the **Create Archive** button to create the Archive. This creates an Archive that is viewable from the **Archives List Page** for the specific Project.

Creating a New Archive keeps a copy (or record) of both the Project data itself along with the PACT source version. This is not usable outside the tool.

When you are finished with your Project, and want to Archive it to meet airworthiness or other certification requirements, if you remain on PACT support, you will have ongoing access to all of your Projects and data, which will be updated to work on the latest version of PACT.

If you need an Archive but do not need to continue using PACT for any other projects, you will need to purchase a *Project Archive license*. This means you take an Archive of your Project and ACS maintains this Archive on our servers, storing it long-term. At your direction at some point in the future, ACS would load the Project and tool so that you could work on the Project again if needed.



NOTE, the Project Archive license is a cost option based on current data storage prices and length of time required to store the Archive. Contact info@airworthinesscert.com for a quote.

13.4 Onsite Use and Archiving

If for security or archival purposes, your organization wants to own a local copy of PACT, (installed on your local server, with minimal maintenance supplied by ACS), this can be achieved. Contact info@airworthinesscert.com if you would like to discuss this special option.

Appendix A: Acronyms

Acronym	Stands For
AC	Advisory Circular
ACS	Airworthiness Certification Services
AI	Action Item
ARP	Aerospace Recommended Practice
AMC	Acceptable Means of Compliance
ATM	Air Traffic Management
AWS	Amazon Web Service
CAST	Certification Authorities Software Team
CBA	Circuit Board Assembly
CCA	Circuit Card Assembly
CCB	Change Control Board
CC1	Control Category 1
CC2	Control Category 2
CEH	Complex Electronic Hardware
CID	Company Identifier
CM	Configuration Management
CNS	Communication, Navigation and Surveillance
CMP	Configuration Management Plan
CRC	Cyclic redundancy check
DAL	Development Assurance Level (formerly, Design Assurance Level)
DO	Document
EASA	European Aviation Safety Agency
ECM	Engineering Coordination Memo
EUROCAE	European Organization for Civil Aviation Equipment
FAA	Federal Aviation Administration
FCI	Firmware Configuration Index

FPGA	Field Programmable Gate Array
HAS	Hardware Accomplishment Summary
HC1	Hardware Change Control Category 1
HC2	Hardware Change Control Category 2
HCS	Hardware Code Standards
HCI	Hardware Configuration Index
HCM	Hardware Configuration Management
HCMP	Hardware Configuration Management Plan
HDD	Hardware Design Document
HDL	Hardware Description Language
HDP	Hardware Development Plan
HDS	Hardware Design Standards
HPA	Hardware Process Assurance
HPAP	Hardware Process Assurance Plan
HRD	Hardware Requirements Document
HRS	Hardware Requirements Standards
HTM	Hardware Traceability Matrix
HTP	Hardware Test Procedures
HTR	Hardware Test Results
HVVP (or HVP)	Hardware Validation and Verification Plan
HVVS (or HVS)	Hardware Validation and Verification Standard
HW	Hardware
HWCI	Hardware Configuration Item
ID	Identifier
IDAL	Item Development Assurance Level
ITAR	International Traffic in Arms Regulations
I/O	Input/Output
MC/DC	Modified Condition Decision Coverage
MBDS	Model-Based Design Standards

OO	Object Oriented
OPR	Open Problem Report
PCB	Printed Circuit Board
PDI	Parameter Data Items
PHAC	Plan for Hardware Aspects of Certification
PLD	Programmable Logic Device
PMC	Program Management Committee
PR	Problem Report
PSAC	Plan for Software Aspects of Certification
RAID	Redundant Array of Independent Disks
RTM	Requirements Traceability Matrix
RTCA	Radio Technical Commission for Aeronautics (now simply "RTCA")
SAE	Society of Automotive Engineers
SAS	Software Accomplishment Summary
SC	Special Committee
SCI	Software Configuration Index
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SCS	Software Code Standards
SDD	System Design Document
SDP	Software Development Plan
SDS	Software Design Standards
SECI	Software Environment Lifecycle Index
SHA	Secure Hash Algorithm
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SRD	System Requirements Document
SRS	Software Requirements Standards
SSH	Secure Shell

SSL	Secure Sockets Layer
STM	Software Traceability Matrix
SVP	Software Verification Plan
SVR	Software Verification Results
SW	Software
SWDD	Software Design Document
SWRD	Software Requirements Document
SVP	Software Verification Plan
SVVP	Software Validation and Verification Plan
SWVCP	Software Verification Cases and Procedures
TBD	To Be Determined
TOR	Terms of Reference
TQL	Tool Qualification Level
TQP	Tool Qualification Plan
TQR	Tool Qualification Results
URL	Uniform Resource Locator
USA (or US)	United States of America
WG	Working Group
VPX	ANSI/VITA 46.0-2007 Standard

Appendix B: Glossary of Terms

Term	Meaning
Activity	Tasks that provide a means of meeting the <i>Objectives</i> .
Administrator	A PACT user with special privileges to set up (and potentially modify) the organization's use of the tool.
Airworthiness Certification Services, LLC.	ACS, the company that provides PACT, was founded and incorporated as a Women Owned Business by President, Tammy Reeve, in January 2017. https://www.airworthinesscert.com/
Archive	A means capture life cycle data associated with a Hardware or Software Item, which can be retrieved in case of a need to duplicate, regenerate, retest or modify the product.
Baseline	The approved, recorded configuration of one or more configuration items, that thereafter serves as the basis for further development, and that is changed only through change control procedures.
Baselines and Traceability	A means of identifying a controlled configuration item as a baseline which means it or a collection of items can have an associated Identification and that specific controlled version of the data is traceability to its previous versions (in other words, you can tell what changed between the baselines and the progression of things.)
Branch	A collection of a set of changes in items. Branching, in version control and software configuration management, is the duplication of an object under version control (such as a source code file or a directory tree) so that modifications can occur in parallel along multiple branches. Branches are also known as trees.
Certification Representative	This is a generic term that can refer to any person or entity who may be charged with verifying compliance to a certification standard.
Change Control	The control data in the configuration management aspects of the tool to ensure that two items don't have the same name but different content

	and that there is a unique identifier for all items control in the configuration management repository
Change Control Board	A group of subject matter experts who make decisions regarding whether or not proposed changes should be made to the project, item or product.
Change Impact Analysis	Identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change.
Change Review	The verification aspects related to updates to versions of documents and actions from Reviews being closed out
Code (or Source Code)	Computer Software (usually as written in its uncompiled text form, using a language such as C++) or a Hardware Description Language (HDL) such as VHDL or Verilog.
Component Level Configuration Item (CLCI)	An abstraction which represents an allocation of system functions to a component of the system implementation such as software, hardware card, FPGA, ASIC or PLD. This abstraction for a component level configuration item allows the user to adapt to industry standards for compliance at the appropriate level for a CLCI.
Conceptual Design(s)	This term relates to DO-254 and refers to one or more a high-level design concepts that may be assessed to determine the potential for the resulting design implementation to meet the Item's Requirements. Note that the equivalent in DO-178C is Low-Level Requirements.
Configuration Identification	Unique numbers or identifiers for configuration items with revision identification.
Configuration Item	A component of a system that can be identified as a self-contained unit for purposes of change control and identification. In other words, version numbers and configuration item registration codes help in uniquely identifying configuration items. DO-178C Definition: 1) One or more hardware or software components treated as a unit for configuration management purpose, (2) software life cycle data treated as a unit for configuration management purposes.

	DO-254 Definition: One or more components, tools or data items treated as a unit for configuration management purposes.
Configuration Management	The process of (a) identifying and defining the configuration items of the system; (b) controlling the release and change of these items through the software lifecycle; (c) recording and reporting the status of configuration items and Problem Reports; and (d) verifying the completeness and correctness of configuration items.
Configuration Status Accounting	The recording and reporting of the information necessary to manage a configuration effectively, including a listing of the approved configuration identification, the status of proposed changes to the configuration, and the implementation of status of the approved changes.
Coverage Analysis	See <i>Test Coverage Analysis</i> , <i>Requirements-Based Coverage Analysis</i> , and <i>Structural Coverage Analysis</i> .
Derived Requirement	A requirement resulting from the hardware/software design processes, which may not be directly traceable to higher level requirements.
Elemental Analysis	The hardware term to describe <i>test coverage analysis</i> , whereby the test cases are analyzed to determine the hardware code elements that they exercise.
Function	A Function is a named section of a program that performs a specific task. This term is used in PACT in the context of handling Source Code.
Hardware Item	An Item (see Item definition that follows), that is a self-contained Hardware Component, which is part of a System.
High-Level Requirement (HLR)	Software requirements developed from analysis of system requirements, safety-related requirements, and system architecture.
Inlinked	When lower level requirements link up to higher level requirements, such as software High Level Requirements linking to System Requirements.

Item	(1) Anything you produce as an artifact, (2) In the system context, an item may refer to Hardware, Software, PCBs, etc. (see CLCI)
(Hardware or Software Item) Life Cycle	(1) An ordered collection of processes determined by an organization to be sufficient and adequate to produce a software product. (2) The period of time that begins with the decision to produce or modify a software/hardware product and ends when the product is retired from service.
Low-Level Requirement (LLR)	Software requirements developed from high-level requirements, derived requirements, and design constraints from which Source Code can be directly implemented without further information.
Module	A module is a software component or part of a program that contains one or more routines. One or more independently developed modules make up a program. This term is used in PACT in the context of handling <i>Source Code</i> .
Objectives	Under DO-254 or DO-178C compliance, objectives are requirements that must be met to demonstrate compliance.
Outlinked	When a System Requirement has a software High-Level Requirement or hardware Requirement associated with it.
Peer	A knowledgeable team member, ideally someone other than the person who is responsible for the authoring of the item under review.
Peer Review	As part of the Verification & Validation process, this type of review examines an Item's documentation and/or artifacts to ensure they comply with compliance expectations and plans.
Problem Reporting, Tracking and Corrective Action	A means of writing up changes due to new development or problems on data controlled in the tool and correlating this issue with the controlled item in the CM aspects of the tool specific to the version of the item the issue or change needs to be corrected in and then the version that the item was fixed ultimately.
Product	An object a company develops, produces and sells, such as a hardware and/or software article. In the PACT context, when you are creating Problem Reports for a

	Hardware or Software Item, you can specify the Product that this Item is related to. In this way, you can sort PRs by Product to see which PRs affect which Product.
Project	A piece of planned work or an activity that is finished over a period of time and intended to achieve a particular purpose. In the context of PACT, the Project is the scope of work defined by the allocated system requirements
Record	A record (used in the context of Source Code and Test Case objects) is a collection of data, captured from fields within the tool interface, which identifies the object. Note that PACT does not manage the object itself, but rather information about the object, which is stored in the record.
Repository	A repository is used to manage project data including application code versions, documents, and tests, and to securely store them from uncontrolled changes or deletion.
Requirement	A requirement is a singular documented physical or functional need that a particular design, product or process aims to satisfy. In the PACT context, as well as that of DO-254 and DO-178C, there are system level requirements, high and low-level requirements for software, and requirements and conceptual designs for hardware, all of which define the requirements for the project and its HW/SW sub-components.
Requirements-Based Coverage Analysis	The process of analyzing the Item's test cases in relation to the requirements to confirm that they satisfy the specified criteria.
Software Item	An Item (see Item definition above), that is a self-contained Software Component, which is part of a System.
Source Code	See also <i>code</i> . 1) For Software: Code written in source languages, such as assembly language and/or high-level language, in a machine-readable form for input to an assembler or a compiler. 2) For Hardware: Any specialized computer language used to describe the structure and behavior of electronic circuits.

	In the PACT context, this term applies to both Software and Hardware Items.
Structural Coverage Analysis	This activity evaluates code structure, including interfaces, exercised during requirements-based testing. For software, this includes examining both data coupling and control coupling.
System	(1) A collection of hardware and software components organized to accomplish a specific function or set of functions. (2) A combination of inter-related items arranged to perform a specific Function(s). Note that in PACT, a System is represented as a "Project."
Test Case	A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.
Test Coverage Analysis	Test coverage analysis is a two-step process involving <i>Requirements-Based Coverage Analysis</i> and <i>Structural Coverage Analysis</i> . For hardware this is called <i>elemental analysis</i> .
Test Procedure	Detailed instructions for the set-up and execution of a given set of test cases, and instructions for the evaluation of results of executing the test cases.
Transition Criteria	The minimum conditions, as defined by the software/hardware planning process, to be satisfied to enter a life cycle process.
Transition Review	As part of the Verification & Validation process, this type of review examines the objectives met and activities performed during an Item's life cycle phase to ensure they have met the criteria in the project plans, and ensures that all Transition Criteria is met before moving on to the next phase.

Appendix C: Roles & Access

The following list identifies, by role, the type of PACT data access each team member has.

- **Administrator**
 - Action Items: List and Show
 - Checklists: List and Show
 - Documents: List and Show both Documents and Comments, Download Documents
 - GitHub/GitLab access: List, Show, Set Repository/Branch/Folder, Get Repositories/Branches/Folders/Files
 - Items: List, Show and Filter
 - Problem Reports: List and Show (including History), and ability to Open
 - Projects: List, Show, Filter and Review Status.
 - Requirements: List, Show and Export (including Traceability Matrices)
 - Reviews: List, Show and Sign-in
 - Source Code: List and Show
 - Templates (Checklists and Documents): Setup of Organization's Template Set; List and Show for Projects
 - Test Cases/Procedures: List and Show
 - User: Full access
- **Project Manager**
 - Action Items: Full access to all operations
 - Checklists: Full access to all operations
 - Documents: Full access to all operations
 - GitHub/GitLab access: Full access to all operations
 - Items: Full access to all operations
 - Problem Reports: Full access to all operations
 - Projects: Full access to all operations
 - Requirements: Full access to all operations
 - Reviews: Full access to all operations
 - Source Code: Full access to all operations
 - Templates (Checklists and Documents): Setup of Organization's Template Set; full access to all operations, including Update and Modify
 - Test Cases/Procedures: Full access to all operations
 - User: List, Show, Create, Update and Delete
- **Configuration Manager**
 - Action Items: Full access to all operations

- Checklists: Full access to all operations
 - Documents: Full access to all operations
 - GitHub/GitLab access: Full access to all operations, with the exception of Update and Delete repositories.
 - Items: Full access to all operations
 - Problem Reports: Full access to all operations
 - Projects: Full access to all operations
 - Requirements: Full access to all operations
 - Reviews: Full access to all operations
 - Source Code: Full access to all operations
 - Templates (Checklists and Documents): Setup of Organization's Template Set; Full access to all operations, including Update and Modify
 - Test Cases/Procedures: Full access to all operations
 - User: List, Show, Create, Update and Delete
- **Quality Assurance** Action Items: List and Show
 - Action Items: List, Show, Create, and Update
 - Checklists: List and Show
 - Documents: List and Show both Documents and Comments, Download Documents
 - GitHub/GitLab access: List, Show, Set Repository/Branch/Folder, Get Repositories/Branches/Folders/Files
 - Items: List, Show and Filter
 - Problem Reports: Full access to all operations except Delete
 - Projects: List, Show, Filter and Review Status.
 - Requirements: Full access to all operations (including Traceability Matrices) except Delete
 - Reviews: Full access to all operations except Delete
 - Source Code: Full access to all operations except Delete
 - Templates (Checklists and Documents): List and Show
 - Test Cases/Procedures: Full access to all operations except Delete
 - User: No access
 - **Certification Representative**
 - Same as **View Only** access, which is defined below.
 - Includes ability to participate in Reviews.
 - **Team Member**
 - Action Items: List, Show, Create and Update
 - Checklists: List and Show

- Documents: List, Show, Create and Update both Documents and Comments, Download Documents
- GitHub/GitLab access: List, Show, Set Repository/Branch/Folder, Get Repositories/Branches/Folders/Files
- Items: List, Show and Filter
- Problem Reports: List, Show, Create and Update (including History), and ability to Open
- Projects: List, Show, Filter and Review Status.
- Requirements: Full access (including Traceability Matrices)
- Reviews: List, Show, Create, Update and Sign-in
- Source Code: Full access to all operations
- Templates (Checklists and Documents): List and Show
- Test Cases/Procedures: Full access to all operations
- User: No access

Two other categories of PACT Access are:

- **View Only**: This category gives the user List and Show access to all objects, along with Set and Get access to Repositories/Branches/Folders, Open PRs, Document Download and Review sign-in.
- **Restricted View**: Same as View Only. Gives read-only access to limited features.