

FAA National Software Conference, May 2002

FAA Reuse Policy



Software Reuse



Leanna Rierson
May 16, 2002

1



Acronyms

- AC Advisory Circular
- ACO Aircraft Certification Office
- API Application Programmer Interface
- CI Configuration index
- FAA Federal Aviation Administration
- LRU Line Replaceable Unit
- PR Problem Report
- PSAC Plan for Software Aspects of Certification
- RSC Reusable Software Component
- RSCD Reusable Software Component Developer
- RTOS Real-time operating system
- SAS Software Accomplishment Summary
- SCI Software Configuration Index
- SW Software

2

FAA National Software Conference, May 2002

FAA Reuse Policy



Two Notices

- **8110.97**
 - Reuse of software life cycle data
 - Formally known as 8110.reuse
 - Completed February 2002
 - Was “messed up” by editorial process
 - Chapter 12 of Mega SW Order (8110.SW)
- **8110.RSC (draft 10)**
 - Reuse of entire components
 - In and across company boundaries
 - Targeted completion ??? – your guess is as good as mine!!! It’s ready to be signed.

3



Chapter 12 of Mega SW Order (Notice 8110.97)

Entitled: “Approving Reused
Software Life Cycle Data”

4

FAA National Software Conference, May 2002

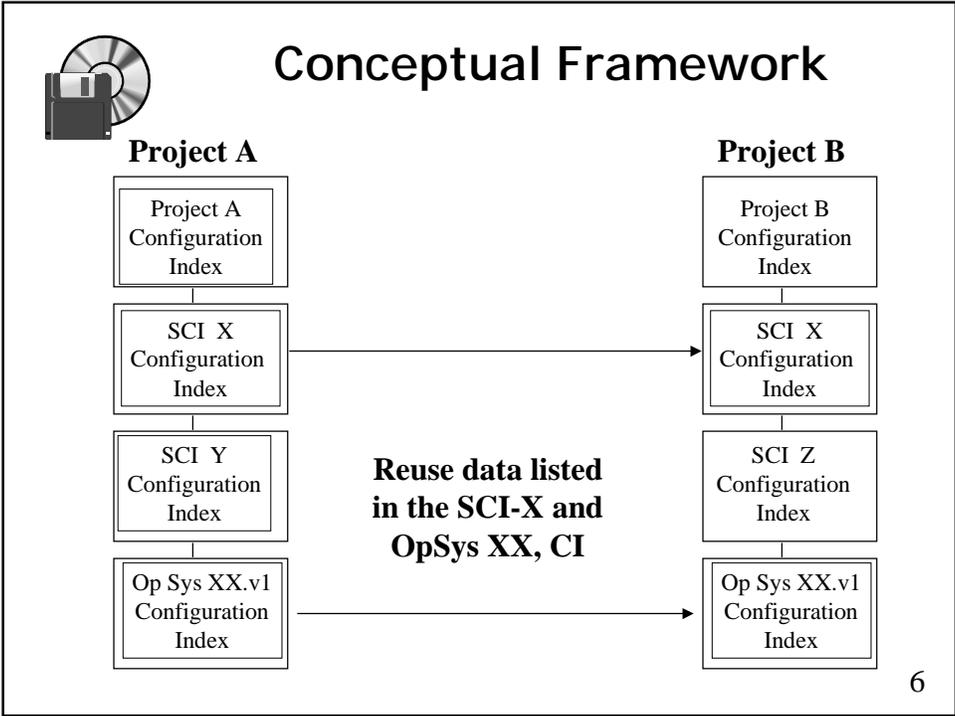
FAA Reuse Policy



Background

- Notice applies to reuse of software life cycle data
- Reusable Components are addressed in a separate notice.
- Good packaging is needed to maximize reuse.

5



FAA National Software Conference, May 2002

FAA Reuse Policy



Examples of Good Packaging

- Develop plans and standards to be as “generic” as possible, with project-specific information in the PSAC.
- Build and package the software so it can be used on multiple projects
- Tool qualification data separate for tools used on all software projects
- Make individual configuration indices (CIs) for components that may later be reused
- Design the software components for reuse (high cohesion, low coupling)

7



Applicable Definitions

(Section 1-7 of 8110.SW)

Original Certification Project	First use of the reusable software life cycle data in a completed cert project.
Subsequent Certification Project	Follow-on project that reuses software life cycle data from the original certification project.
Reuse	Subsequent use of unaffected, previously approved software life cycle data.
Certification Credit	Acceptance that a process, product, or demo meets the certification requirements.

8

FAA National Software Conference, May 2002

FAA Reuse Policy



Applicable Definitions (cont)

(Section 1-7 of 8110.SW)

Software Life Cycle Data	Data produced during the software life cycle. Also known as the DO-178B, Section 11 data.
Configuration Item	1) One or more software components treated as a unit. 2) Software life cycle data treated as a unit.
Software Configuration Index	Identifies configuration of an item. Contains one or more configuration items.
Software Life Cycle Env. Index	Identifies configuration of the software life cycle environment.

9



Applicable Definitions (cont)

(Section 1-7 of 8110.SW)

Software Plans & Standards	Data that directs the development & integral processes.
Software Tool	Computer program used to develop, test, analyze, produce, or modify another program or its documentation.
Tool Qualification	Process necessary to obtain cert credit for a tool.
Software Library	Collection of software and related data/documents.

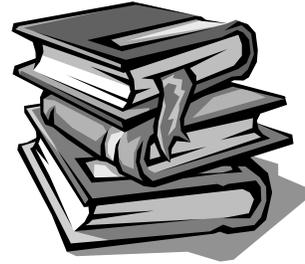
10

FAA National Software Conference, May 2002

FAA Reuse Policy



5 Sections



- 12-1: General
- 12-2: Software Suitable for Reuse
- 12-3: Safety Considerations
- 12-4: Factors Affecting Reuse
- 12-5: Reuse Approval Guidelines

11



12-2: Software Suitable for Reuse

- Software plans and standards
- Tool qualification data
- Software libraries
- Software requirements, design, code, verification procedures, and verification results.
- Configuration items
- *Basically: any unchanged software life cycle data*

12

FAA National Software Conference, May 2002

FAA Reuse Policy



12-3: Safety Considerations

- **FAA can approve for reuse if:**
 - There is no adverse effect on original systems safety margins, **and**
 - There is no adverse effect on original operational capability **UNLESS** accompanied by justifiable increase in safety.

13



12-3: Safety Considerations (cont)

- **FAA will not approve for reuse if reuse:**
 - Adversely affects safety,
 - Exceeds a pre-approved range of data or parameters, or
 - Exceeds equipment performance characteristics.

14

FAA National Software Conference, May 2002

FAA Reuse Policy



12-4: Factors Affecting Reuse

- a. Any Section 11 data can be reused if:
 - It remains unchanged
 - It is applicable to the project
 - No safety issues exist
- b. In-service problems might limit reuse
- c. Open problems reports should be analyzed prior to reuse

15



12-4: Factors Affecting Reuse (cont)

- d. Assessment should be performed to show similarity of operational environment and safety assessment
 - Builds on a and c

16

FAA National Software Conference, May 2002

FAA Reuse Policy



12-5: Reuse Approval Guidelines

- Certification authority should ensure that:
 - Data to be reused is unchanged.
 - The software level is equivalent to (or less than) software level of the previous approval.
 - Range & data type of inputs are equivalent to previous approval.
 - Configuration items are used on the same target environment and in same operational way.

17



12-5: Reuse Approval Guidelines (cont)

- Certification authority should ensure that:
 - Equivalent software/hardware integration and system testing conducted on same target and system as previous approval.
 - Applicant addressed safety considerations.
 - Reuse rationale is documented in "Additional Considerations" portion of the PSAC.

18

FAA National Software Conference, May 2002

FAA Reuse Policy



Summary of Chapter 12 of Order 8110.SW (Notice 8110.97)

- Reuse of software life cycle data on multiple certification projects is feasible
- If a data item hasn't changed and is applicable for the current project, it is a candidate for re-use
- Present plan for reuse in PSAC and get early ACO agreement
- Thanks to Dennis and John!

19



8110.RSC

Entitled:

"Guidelines for Accepting
Reusable Software Components"
(Draft 10)

*Goal: To be able to carry certification "credit"
for reusable software component from one project
to the next*

20

FAA National Software Conference, May 2002

FAA Reuse Policy



11 Sections

- 1-3: Purpose, Distribution, Related Publications
- 4: Definitions
- 5: Acronyms
- 6: Background
- 7: Discussion
- 8: Guidelines
- 9: Common Reuse Issues and Considerations
- 10: Changes to Components
- 11: Conclusion

21



Section 4 Highlights ~ Definitions ~

- Reusable software component (RSC) is the software code and its supporting DO-178B documentation being considered for reuse. It forms a portion of the software that will be implemented by the integrator/applicant.
- Reusable software component developer (RSCD) is the manufacturer of the reusable software component.

22

FAA National Software Conference, May 2002

FAA Reuse Policy



Section 4 Highlights (cont) ~Definitions~

- Integrator is the manufacturer responsible for integrating the re-useable software component into the target computer and with other software components.
- Applicant is the manufacturer seeking certification or authorization of the overall system.

23



Section 6 ~Background~

- Traditionally, software approval is at the system level.
- There is currently no vehicle to carry certification credit across project boundaries.
- Purpose of this notice is to provide guidelines for allowing "credit" across projects.

24

FAA National Software Conference, May 2002

FAA Reuse Policy



Section 6 (cont) ~Background~

- RSC Examples:
 - Operating Systems
 - Libraries
 - Input/Output Data Files
 - Loading Software
- Guidelines are applicable within a company or across company boundaries.

25



Section 7 ~Discussion~

STAKEHOLDERS

Integrator

**Reusable Software
Component Developer
(RSCD)**



Applicant

**Certification
Authorities**

Note: Cert authorities may have more involvement than a traditional software development project for the initial component development

FAA National Software Conference, May 2002

FAA Reuse Policy



Section 7 (cont) ~ Discussion ~

- Scoping the RSC guidelines:
 - 7a – First acceptance of RSC is a “real” project
 - 7b – Stakeholders agree on reuse
 - 7c – Each project is unique and might have different “credit”
 - 7d – Applicant is responsible for final cert

27



Overview of the Process

Stakeholders agree that reuse is a desirable & obtainable goal.

RSCD, integrator, & applicant plan for reuse.

RSCD, integrator, & applicant document reuse credit per objective.

PSAC reviewed & approved by cert authorities.

RSC developed per plans with cert authority oversight.

ACO writes acceptance letter for RSC to RSCD and applicant.

Same configuration & version of RSC used on other programs within limitations.

28

FAA National Software Conference, May 2002

FAA Reuse Policy



Section 8a – Guidelines for RSC Developer

- Document Reuse intent in the PSAC
 - Reuse credit for each objective
 - Assumptions for each objective
 - Means of compliance for each objective
 - Remaining activities for the installer/applicant for each objective

29



Example Approach (Appendix 1)

<i>178B Obj #</i>	<i>Obj Description</i>	<i>Credit Sought</i>	<i>Assumption</i>	<i>Means of Compliance for the Objective</i>	<i>Activities Remaining For Integrator/Applicant</i>
1-1	Software development and integral processes activities are defined.				
1-2	Transition criteria, inter-relationships and sequencing among processes are defined.				

30

FAA National Software Conference, May 2002

FAA Reuse Policy



Example Approach (cont)



- Document objective, credit sought, assumptions, and remaining activities in the PSAC and Accomplishment Summary.
- Address target dependencies.
- Address assumptions regarding requirements; particularly high-level requirements.
- Be specific and thorough.
- Obtain FAA input & agreement on proposals up-front.

31



Example Approach (cont) [Full Credit]

- **Objective 1-1:** Software development and integral processes activities are defined.
- **Credit Sought:** Full
- **Assumptions:** Plans are completed and unchanged for router.
- **Remaining Activities:** Applicant/integrator to complete LRU level plans, reference router plans/data, & consider reuse in "Additional Considerations"

32

FAA National Software Conference, May 2002

FAA Reuse Policy



Example Approach (cont) [Partial Credit]

- **Objective 2-1:** High-level requirements are developed.
- **Credit Sought:** Partial
- **Assumptions:** Assuming high level requirements are document XXX, revision - and the LRU manufacturer uses those requirements.
- **Remaining Activities:** Because the high-level requirements actually exist at the LRU level, they cannot be fully implemented at the software component level. The applicant may reference and tie to the component-level high-level requirements as their own high-level requirements. If this occurred, the applicant would also need to verify the high-level functionality of these requirements in their system.

33



Section 8a (cont) – Other Responsibilities of RSC Developer

- Document safety-related issues
- Coordinate & follow plans with all stakeholders
- Submit SAS and SCI at end of project, with the completed compliance tables
- Supply data to support the type certificate to the applicant

34

FAA National Software Conference, May 2002

FAA Reuse Policy



8b – Integrator/Applicant Responsibilities

- Integrate RSC data into the project data
- Specify the life cycle data needed from the applicant
- Consider safety issues of the RSC
- Coordinate & follow plans
- Consider open PRs of the RSC
- Validate assumptions made by the RSC developer
- Complete the RSC objectives tables in the SAS

35



8c – Cert Authority Guidelines on 1st Approval of the RSC

- Involve all stakeholders
- Involve technical experts, as needed
- Review plans of RSC developer and 1st applicant for consistency
- Perform reviews, as needed
- Approve project, when objectives are satisfied
- Write letter for RSC developer explaining acceptance, limitations, etc.

36

FAA National Software Conference, May 2002

FAA Reuse Policy



8d – Cert Authority Guidelines on Subsequent Use of RSCs

- Review the acceptance letter
- Contact ACO engineer who did the original acceptance, if needed
- Ensure that the applicant follows the guidelines of this notice
- Perform reviews of project plans and data
- Ensure consistency between RSC plans/data and applicant's plans/data
- Inform original ACO of subsequent use/approval of RSC

37



Section 9 – Common Issues & Considerations

- High-level requirements objectives
- Re-verification issues
- Interface documents
- Partitioning/Protection Considerations
- Safety issues
- Coupling & Cohesion

38

FAA National Software Conference, May 2002

FAA Reuse Policy



Section 10 – Changes to RSCs

- When RSC is changed, cannot be reused without another reuse application.
- Changes to RSC may or may not affect all users.

39



What's Next?

- Hopefully, the Notice will be released SOON.
- Several projects are using the concept.
- Collecting lessons learned. E.g.,
 - Dealing with deactivated code
 - Concurrent programs using a component
 - Derivative versions of the RSC
 - Additional items for RTOSes (e.g., board-support packages, APIs, ...)
- Once concept matures a bit more, an Order and/or AC will be created.
- Please send any questions, comments, etc. to Leanna.Rierson@faa.gov

40

FAA National Software Conference, May 2002

FAA Reuse Policy



Summary

- We are making progress in addressing software reuse
- 8110.97 is completed and is being rolled into the mega order
- 8110.RSC should be finished soon
- Tool qualification reuse is to be addressed in future policy

41

Attachment #1
of "Software Reuse"

Chapter 12 of
Order 8110.SW

CHAPTER 12. APPROVING REUSED SOFTWARE LIFE CYCLE DATA

12-1. GENERAL. This chapter provides guidelines for determining if software life cycle data, produced and approved for one certification project, can be approved on a follow-on certification project. Approval for reuse could minimize the amount of rework while maintaining an equivalent level of design assurance.

12-2. SOFTWARE SUITABLE FOR REUSE.

a. If properly planned and packaged, software life cycle data can be reused from one project to the next, with minimal rework. For example, the software plans, requirements, design, and other software life cycle data (as documented in a software configuration index) for a Global Positioning System (GPS) may originally be approved on GPS #1 (the original certification project) and reused on GPS #2 (the subsequent certification project). Sample items suitable for reuse include:

(1) Software plans and standards. These include software undergoing non-substantive changes, such as:

- Program name,
- Name change due to consolidations or mergers, and
- Configuration changes for reasons other than design changes (for example, document format change, drawing modifications, or documentation system changes).

(2) Tool qualification data. The FAA can approve reuse if the tool is used in exactly the same way as the original certification, and the applicant has access to the tool qualification data. The applicant should ensure that the same version of the tool is being used as in the previous developments. The FAA will not approve reuse if the applicant uses additional or different tool functionality than was previously qualified.

(3) Software libraries. The FAA can approve library sets in the original certification project if the library set is used identically (that is, same library functions are used the same way).

(4) Software requirements, design, code, verification procedures, and verification results. The FAA may approve these for reuse after the applicant makes a thorough change impact analysis. This is to confirm that the requirements, design, code, procedures and so forth are unaffected and unchanged from the previous certification effort.

(5) Configuration items. These may be approved for reuse in their entirety, if the certification authority and DERs use sections 12-3 through 12-5 of this chapter to make the determination, and the configuration of the software life cycle data has not changed. Configuration item requirements verified at a higher level (that is, system level) should be identified in the original configuration and re-verified before reuse.

b. Projects not using RCTA/DO-178B may have additional considerations not documented in this chapter. Certification authorities should evaluate them on a case-by-

case basis. The applicant should contact their local certification authority for guidance. The certification authority should coordinate with the CSTA for Aircraft Computer Software, the appropriate Directorate, and/or AIR-120, as necessary.

12-3. SAFETY CONSIDERATIONS. If the FAA finds software life cycle data acceptable for reuse, no further design approval is required. The following table illustrates the considerations that govern whether the FAA will approve software reuse.

Table 12-1. Reuse Approval Considerations

FAA can approve for reuse if:	<ul style="list-style-type: none"> • There is no adverse effect on original systems safety margins, and • There is no adverse effect on original operational capability UNLESS accompanied by justifiable increase in safety.
FAA will NOT approve for reuse if:	<ul style="list-style-type: none"> • It adversely affects safety, • Exceeds a pre-approved range of data or parameters, or • Exceeds equipment performance characteristics.

12-4. FACTORS AFFECTING REUSE.

a. Any of the software life cycle data in Section 11, RTCA/DO-178B is suitable for reuse. To meet the guidelines in section 12-5 of this chapter, the software life cycle data should be unchanged, and should apply to the project for which reuse is being considered.

b. In-service problems with previous applications can limit reuse. There may be Airworthiness Directives or a manufacturer's unresolved problem reports with the previously approved system. The applicant should either carefully evaluate the reuse of software life cycle data of that product, or not use it.

c. The applicant needs to analyze all open manufacturer's problem reports to ensure that the reusable portion of the new software is not affected. If the reusable portion of the new software is affected, make changes to correct that software life cycle data.

d. Applicants should determine if the software data applies to the subsequent project's development by assessing the similarity of both the operational environment, and the software development environment. They should:

- Assess the operational environment by evaluating the end-to-end performance requirements and the operational safety assessment.
- Refer to the Software Life Cycle Environment Configuration Index in section 11.15, RCTA/DO-178B, when assessing the software development environment.
- Demonstrate that operational and development environments are the same, or demonstrated to produce identical results as the previous certification.
- Assess any outstanding problem reports.

12-5. REUSE APPROVAL GUIDELINES.

a. The certification authority should ensure that the applicant has met the following guidelines before granting certification credit for reused software life cycle data:

(1) The software life cycle data has not changed since its previous approval.

(2) The software level of the software application(s) is equal to, or less, than the software level of the original certification effort.

(3) The range and data type of inputs to the configuration item are equivalent to its approved predecessor.

(4) The configuration item is embedded on the same target computer and is used the same way operationally as the original certification project.

(5) Equivalent software/hardware integration testing and system testing were conducted on the target computer and system as in the original certification project.

(6) The applicant followed the safety considerations and reuse factors in sections 12-3 and 12-4 of this chapter.

(7) The software life cycle data and the rationale for reuse of each item is documented in the “Additional Considerations” portion of the PSAC. The applicant’s PSAC should include method of use, integration, and documentation for the reused configuration item. The PSAC should be submitted as early as possible in the development program. The applicant documents all references to the project previously certified and the project number, as applicable.

b. The certification authority responsible for the subsequent certification should review the PSAC and notify the applicant whether the proposal is acceptable or not, with a rationale.

Attachment #2
of "Software Reuse"

Notice 8110.RSC
Draft 10

NOTICE

U.S. Department of Transportation
Federal Aviation Administration

N 8110.RSC

DRAFT 10 - April 18, 2002

Cancellation

Subject: GUIDELINES FOR ACCEPTING REUSABLE SOFTWARE COMPONENTS

1. PURPOSE. This notice guides Aircraft Certification Office (ACO) engineers and Designated Engineering Representatives (DER) who accept software components that may comprise only a portion of an overall system's software application.

a. Advisory Circular (AC) 20-115B implements RTCA/DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*, as a means for obtaining software approval. This notice is intended to provide guidelines for projects that use RTCA/DO-178B as a means of compliance. If an applicant proposes a different means of compliance than RTCA/DO-178B, the Federal Aviation Administration (FAA) should decide whether this notice applies and if additional policy is warranted.

b. This notice applies to the approval of airborne systems and equipment and the software aspects of those systems related to type certificates (TC), supplemental type certificates (STC), amended supplemental type certificates (ASTC), amended type certificates (ATC), and Technical Standard Order (TSO) Authorizations.

c. This notice supplements RTCA/DO-178B and AC 20-115B, for approving some or all of the RTCA/DO-178B objectives for individual components of a system's software application.

2. DISTRIBUTION. This notice is distributed to the branch level in Washington Headquarters Aircraft Certification Service, section level in all Aircraft Certification Directorates, all Chief Scientific and Technical Advisors (CSTA), all Aircraft Certification Offices (ACO), all Manufacturing Inspection Offices (MIO), Certificate Management Office (CMO), all Manufacturing Inspection District or Satellite Offices (MIDO/MISO), all Certificate Management Units (CMU), all Flight Standards District Offices (FSDO), and Designated Engineering Representatives (DER). Additional limited distribution should be made to the Air Carrier District Offices, the Aeronautical Quality Assurance Field Offices, and the FAA Academy.

3. RELATED PUBLICATIONS.

a. AC 20-115B, *RTCA, Inc. Document RTCA/DO-178B*, dated January 11, 1993.

b. RTCA, Incorporated, document RTCA/DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*, dated December 1, 1992.

c. FAA Notice 8110.85, *Guidelines for the Oversight of Software Change Impact Analyses Used to Classify Software Changes as Major or Minor*, dated May 11, 2000.

d. FAA Order 8110.4B, *Type Certification Process*, dated April 24, 2000.

e. Title 14 Code of Federal Regulations (14 CFR), Part 21, *Certification Procedures for Products and Parts*.

f. FAA Notice 8110.97, *Guidelines for Approving Reused Software Life Cycle Data*, dated February 5, 2002.

g. RTCA, Incorporated, document RTCA/DO-248B, *Final Report for Clarification of DO-178B Software Considerations in Airborne Systems and Equipment Certification*, dated October 12, 2001.

h. FAA Notice 8110.91, *Guidelines for the Qualification of Software Tools Using RTCA/DO-178B*, dated January 16, 2001.

4. **DEFINITIONS.** For purposes of this notice, the RTCA/DO-178B Annex B definitions and the following definitions apply:

a. **Access mechanism** is the manner in which a software component is called upon to perform its intended function. This includes invocation mechanisms and data flow to and from the component. This is typically part of the interface control document.

b. **Applicant** is the manufacturer seeking certification of the product or authorization of the equipment.

c. **Certification Authority** is the organization or person responsible within the state or country concerned with the certification of compliance with the requirements.

d. **Credit** is the compliance to one or more RTCA/DO-178B objectives supported by RTCA/DO-178B software life cycle data. This compliance is used to show that the certification basis has been met and the equipment may receive a certificate. Three types of credit are referred to throughout this notice:

(1) Full credit – fully meets the RTCA/DO-178B objective and requires no further activity by the user.

(2) Partial credit – partially meets the RTCA/DO-178B objective and requires additional activity by the user to complete compliance.

(3) No credit – does not meet the RTCA/DO-178B objective and must be completed by the user for compliance.

e. Derived high-level requirements are the highest level software requirements that the reusable software component (RSC) developer uses to design and implement his software. They are being called derived because they are not traceable to any higher level requirements by the RSC developer. They may be traceable by the integrator of the RSC to requirements of his system or software. Calling these requirements derived means they should be reviewed by the integrator's systems group for safety implications.

f. Documentation configuration is the numbering used to identify the configuration of documents used in the development process.

g. Integrator is the manufacturer responsible for integrating the reusable software component into the target computer and system with other software components.

h. Installation procedures are procedures used to install the RSC. These might be documented in the porting guide, interface control document, or similar document.

i. Interface control document is used to identify the interface details of the RSC. It is provided by the RSC developer for use by the integrator and applicant. The interface control document should explicitly define what activities are required by the integrator and/or applicant to ensure that the RSC will function in accordance with its approval basis.

j. Maintenance code is code residing in a line replaceable unit (LRU) that interfaces with an onboard maintenance computer or computer used by maintenance personnel. The function of this code is usually to report to the maintenance computer any problems detected during normal operations.

k. Porting guide is a guide that documents assumptions and limitations on the reuse of the component that must be followed to ensure correct functioning of the component in a new environment.

l. Reusable software component (RSC) is the software, its supporting RTCA/DO-178B software life cycle data, and additional supporting documentation being considered for reuse. The component designated for reuse may be any collection of software, such as, libraries, operating systems, or specific system software functions.

m. RSC developer is the manufacturer of the RSC.

n. Settable parameters are software component data that are set before execution of the component.

o. Software component is some part of the LRU software. It is usually defined as performing specific functions within the LRU.

p. Software life cycle data is data produced during the software life cycle to plan, direct, explain, define, record, or provide evidence of activities (see RTCA/DO-178B, Section 11.0).

Sections 11.1 through 11.20 of RTCA/DO-178B describe different kinds of software life cycle data.

q. Stakeholders are all the entities involved in the development, integration, and acceptance of the RSC. Stakeholders mentioned in this notice are the RSC developer, integrator, applicant, and certification authority. The roles of the RSC developer, integrator, and applicant may be assumed by one or more manufacturers.

r. Target computer is the physical processor that will execute the program while airborne.

s. Target computer environment is the target computer and all its support hardware and systems needed to function in its actual airborne environment.

t. Target environment is the same as target computer environment (above).

u. Variables are named memory locations that contain data that will change during software execution.

5. ACRONYMS. The following is a list of acronyms used in this notice.

AC	Advisory Circular
ACO	Aircraft Certification Office
AD	Airworthiness Directive
ASTC	Amended Supplemental Type Certificate
ATC	Amended Type Certificate
CFR	Code of Federal Regulations
CI	Configuration Index
CMO	Certificate Management Office
CMU	Certificate Management Unit
CSCI	Computer Software Configuration Item
CSTA	Chief Scientific and Technical Advisor
CSU	Computer Software Unit
DER	Designated Engineering Representative
FAA	Federal Aviation Administration
FMS	Flight Management System
FSDO	Flight Standards District Office
LRU	Line Replaceable Unit
MC/DC	Modified Condition / Decision Coverage
MIDO	Manufacturing Inspection District Office
MIO	Manufacturing Inspection Office
MISO	Manufacturing Inspection Satellite Office
PSAC	Plan For Software Aspects Of Certification
RSC	Reusable Software Component
SAS	Software Accomplishment Summary
SCI	Software Configuration Index

SCMP	Software Configuration Management Plan
SDP	Software Development Plan
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SSA	System Safety Assessment
STC	Supplemental Type Certificate
SVP	Software Verification Plan
SW	Software
TBD	To Be Determined
TC	Type Certificate
TSO	Technical Standard Order

6. BACKGROUND.

a. Software in airborne systems and equipment is typically approved at the system level. For example, the software for a flight management system (FMS) or electronic engine control is approved with the system, target computer, and target environment fully identified. The target environment is identified up front and the software is developed, integrated, and verified on the target computer.

b. Economic motivations and technical advances in software development have made it desirable to develop a RSC that can later be integrated into a number of target computers and/or environments, as the integrator and/or applicant determine most appropriate for their application needs. In these cases, the developer of the RSC may meet some of the RTCA/DO-178B objectives, while the integrator and/or applicant will be responsible for completing the software and certification compliance activities. Examples of potential RSCs include software libraries, input/output data files, operating systems, and routers.

c. Currently, there are no procedures for RSC developers to directly transfer their accepted data from one project to the next. Traditionally, RSC developers are requested by the FAA to provide substantiation in one of two ways: (1) To resubmit the data package and/or to repeat the work for each system's application. (2) To provide traceability through the TC, ATC, STC, ASTC, or TSO approval back to the desired data and defend the validity of the original objectives approval basis for each application.

d. This notice provides guidelines for certification authorities to grant "credit" for full or partial compliance to RTCA/DO-178B objectives for a RSC. The "credit" is granted through an acceptance letter on the first approval of the RSC. If the RSC is unchanged, that acceptance letter can be presented on additional projects in order to minimize the amount of rework by the RSC developer and FAA. This notice assumes that the RSC being considered for acceptance has its own set of software life cycle data.

e. It should be noted that there are likely other approaches to addressing acceptance of RSCs than the approach described in this notice. Such approaches should be evaluated by the appropriate FAA software experts and additional policy may be needed.

f. Notice 8110.97 was completed by the FAA in February 2002. 8110.97 addresses the reuse of software life cycle data that is used within a company. This notice, on the other hand, focuses on the acceptance of RSCs that may not meet all RTCA/DO-178B objectives and that may be used across company boundaries.

7. DISCUSSION.

a. The first acceptance of a software component must be performed during an actual project (i.e., a TC, ATC, STC, ASTC, or TSO project). This may require more resources for the software component developers, the integrator, the applicant, and the certification authority. Subsequent acceptance will likely require less effort and resources, if the guidelines in this notice are followed.

b. These reuse guidelines apply only when the applicant, integrator, RSC developer, and certification authority (i.e., all the stakeholders) agree that the software component is reusable. The RSC Plan for Software Aspects of Certification (PSAC) and the system-level PSAC are typically the vehicle for agreement and communication among stakeholders. This agreement on the reuse concept is important, because the first applicant will likely use additional resources to make the component reusable. If the reuse concept is not agreed upon by the stakeholders, then the applicant may use one of the traditional approaches mentioned in section 6.c of this notice or seek additional guidance from the FAA.

c. Each RSC developer's project may have different limitations and issues to be addressed. For example, one developer may package its project so it fully meets a particular RTCA/DO-178B Annex A objective, while another developer may only partially meet that same objective. This may be due in part to some project-specific issues and/or additional coordination with the integrator to augment the work of the RSC developer. Section 8 of this notice provides guidelines for addressing reuse of software components: for the RSC developer, the integrator and/or applicant, and the certification authority and/or designee, if delegated. The guidelines are intended to be flexible enough to meet the needs of the RSC developer, integrator, and applicant, but detailed enough to ensure that all issues pertaining to certification are addressed.

d. The applicant is responsible for submitting compliance data and following the certification liaison process for the project. However, there will also likely be communication between the certification authority and the RSC developer for the reuse aspects of the program.

8. GUIDELINES. This section provides guidelines for each of the stakeholders involved in the development and acceptance of a RSC. While applying these guidelines, the common reuse issues address in section 9 of this notice should be consider.

a. **The RSC developer.** The RSC developer should consider the following guidelines when seeking initial acceptance of their software component:

(1) The RSC developer should produce a PSAC for the RSC as early as possible in the project. The PSAC should include the information outlined in Section 11.1 of RTCA/DO-178B. The PSAC should detail the RSC developer's plans for satisfying each applicable RTCA/DO-178B objective, identifying which objectives will not be satisfied, and identifying which objectives will be partially satisfied. For each applicable RTCA/DO-178B objective, the RSC developer should document the following information (either directly in the PSAC or by reference to other documents) with sufficient detail for certification authority concurrence and use by any integrator or applicant who will use the RSC:

(a) **Credit being sought for the objective.** The PSAC or referenced document should specify if full, partial, or no credit is being sought for the objective.

(b) **Assumptions of the RSC developer on the behavior of the RSC users.** Provide sufficient justification to ensure that if the assumptions are satisfied, the original acceptance will be valid. Some examples of assumptions are:

1. The source code is not changed.
2. A means of providing invocation of the component every TBD (to be determined) seconds is provided in order to address timing and resource limitations.
3. Access to the data in interface control drawing for the RSC is provided in the specified format.
4. Enough contiguous physical or virtual address space is provided for the program and data.
5. No calls are allowed to any routines not listed in the specified applications guide.
6. Required software level is not increased.

(c) **Means of compliance for the objective.** The PSAC should document what data will be provided to support compliance to each objective. This should include document names where possible or a description of the type of data to be provided.

(d) **Activities remaining for the integrator and/or applicant.** The PSAC should document what needs to be done by the applicant and/or integrator to fully satisfy any given partial or unsatisfied objective.

1. The applicant may include this information in a table format with columns for the objective number, objective description, credit being sought, assumptions, means of compliance, and remaining activities. An example format is shown in Appendix 1.

2. Since target computer-specific and system-specific issues may be uncertain early in the project, the PSAC may list preliminary information that will require updating in revisions

to the PSAC and/or the Software Accomplishment Summary (SAS). Some reuse details may not be finalized until the end of the project.

(2) The following safety-related items should be documented in the RSC PSAC and RSC SAS:

(a) The software level for the RSC,

(b) Any safety objectives or safety-related requirements allocated to the RSC,

(c) Known effects of the RSC to the safety assessment, and

(d) Architectural and design features supporting any portion of the safety analysis or partitioning or other protection strategies.

(3) The PSAC and any other plans (e.g., Software Development Plan (SDP), Software Verification Plan (SVP), Software Quality Assurance Plan (SQAP), Software Configuration Management Plan (SCMP)) should be coordinated with the certification authority, designees (if delegated), and the applicant and/or integrator for the first application to ensure acceptance by all stakeholders.

(4) Once the RSC developer has agreement on the PSAC and other plans by the stakeholders, they should develop the RSC in compliance with those plans. As previously stated, the RSC developer should produce the RTCA/DO-178B software life cycle data and documentation identified in this policy for the RSC (e.g., plans, standards, requirements, quality assurance records, configuration management records) so that the RSC's life cycle data receives an acceptance by the certification authority (as will be described in sections 8.c and 8.d of this notice).

(5) The RSC developer should inform the certification authority, designees (if delegated), integrator, and applicant of both development progress and any deviations from plans, so reviews and adjustments may be performed in a timely manner.

(6) When the RSC developer is done, the Software Configuration Index (SCI) and the SAS should be submitted to the certification authority through the project integrator, applicant, and/or designee, if delegated. The SAS should include or refer to the software life cycle data of RTCA/DO-178B, Section 11, as well as the information included in section 8.a.(1) of this Notice.

(7) The RSC developer should supply the appropriate software life cycle data to the integrator and/or applicant to support certification of the system in which the RSC will be used. Typically, the following data should be supplied by the RSC developer to both the RSC integrator and applicant:

(a) The type design data listed in Section 9.4 of RTCA/DO-178B for the RSC (i.e., Software Requirements Data, Design Description, Source Code, Executable Object Code, SCI, and SAS).

(b) The PSAC, which lists the credit being sought for each RTCA/DO-178B objective.

(c) Interface information (e.g., interface control document, porting guide). Include any hardware and software resource requirements (e.g., timing and memory requirements).

(d) Installation or integration procedures and limitations. The procedures and limitations should be sufficient to ensure that the RSC meets the requirements. The procedures and limitations should be detailed enough to identify unique aspects of the installation or integration. The limitations and procedures should include, as a minimum:

1. Equipment specifications required for proper operation and performance of the RSC.
2. A list of any sub-components (by part number and version number) that make up the RSC.
3. Instructions for periodic maintenance and calibration needed for continued airworthiness once the software is installed.

(e) Data to support the integrator's and/or applicant's completion of partially satisfied or unsatisfied objectives. As an example, if partial credit was sought for objective 1-1 (Software development and integral processes are defined), it needs to be clearly defined to the integrator and/or applicant what that partial credit entails and what they need to do to complete the credit for the installation. The necessary data to support that "partial" credit should also be made available to the integrator and/or applicant.

Note: Another example of information provided to the integrator and/or applicant is a requirement to perform specific testing on the target environment to completely satisfy specific objectives.

(f) Test cases and procedures to be re-executed on the target environment. This should include a list of test cases and procedures affected by any integrator and/or applicant settable parameters. The integrator and/or applicant need to consider the total requirements for system and sub-system testing, taking credit for reusable tests, re-testing where new settings affect the requirements and code, and producing new test cases and procedures in order to complete all test objectives.

(g) Summary of open problem reports on the RSC and analysis of the operational and safety effects. (Note: This information should be documented in the RSC SAS and may be included in the RSC PSAC, if the information is known up front.)

(h) The following items should also be considered:

1. Any RTCA/DO-178B software life cycle data not listed above used in the software development and approval process should be made available to the applicant, integrator and certification authority (e.g., Software Quality Assurance (SQA) records, Tool Qualification Data).

2. Irrespective of any legal and proprietary issues and agreements about the delivery of software life cycle data between the applicant and the RSC developer, the data should be available to the FAA at all times for their review and inspection. A process may be set up to make the data available to the applicant without actually supplying the data to the applicant (e.g., a data/software escrow). This data should be accessible to the certification authority to determine compliance, or in the event of safety problems with the target system (reference 14 CFR 21.277). The data may also need to be available to the applicant, if the target system or RSC needs to be modified (reference 14 CFR part 21.301 – 21.305, and FAA Order 8110.4B).

3. Data needed to support changes to the RSC should be identified. For example, if the developer should go out of business, this data will help support continued airworthiness. 14 CFR part 21 requires, and FAA Order 8110.4B (chapters 2 and 3) provides guidance, on the preservation of type design data for continued airworthiness of products in the event that the product requires modification because of a safety issue, or if a company goes out of business. Although outside the scope of this notice, RSC developers should ensure compliance with these documents.

4. The RSC developer may find it useful to develop a data sheet for the RSC. This data sheet might summarize RSC functions, limitations, safety concerns, assumptions, configuration, supporting data, etc. in a concise manner and may be included as part of the FAA acceptance letter.

(8) The RSC developer should keep a list of all aviation customers buying or using their components to support continued airworthiness across multiple products. (Note: It is also recommended that the RSC developer and users set up a process to share problem reports in a timely manner.)

b. The integrator and/or applicant using the RSC. In many cases, the integrator and applicant are the same entity. Therefore, their guidelines are addressed together in this subsection. The integrator and applicant should work closely together to meet the following guidelines:

(1) Integrate into their data submittals the RSC developer's plans, documentation, limitations, compliance statement and matrix with RTCA/DO-178B objectives, software approval approach, etc.

(2) Specify the RTCA/DO-178B software life cycle data needed from the RSC developer to support their project and continued airworthiness. The typical software life cycle data needed to support certification is listed in section 8.a.(7) above.

(3) Produce a PSAC (and/or equivalent system-level certification plan) for the target system including the information outlined in RTCA/DO-178B section 11.1. Additionally, the PSAC should include the integrator's and applicant's plans to address compliance with all RTCA/DO-178B objectives for the RSC and other software components of the target system.

(4) Produce other software plans (i.e., SDP, SCMP, SVP, and SQAP) for their target system. Each plan should address the RSC integration, as appropriate, and other software components used. For example, the SVP should address the overall software verification program, as well as any verification required for integration of the RSC and other components, and the credit proposed for the RSC developer's verification.

(5) Consider the safety issues identified in the RSC developer's PSAC, SAS, and/or safety assessment; determine the applicability of those safety issues to the specific application; determine any additional safety issues for the specific application; and address all safety issues.

(6) Coordinate the PSAC and other plans (as needed) with the certification authority and designees (if delegated) to get concurrence on the project.

(7) Follow the approved plans.

(8) Analyze any open problem reports on the RSC to ensure that there are no safety or operational effects from the RSC in the specific application.

(9) Validate the assumptions and claims for credit made by the RSC developer.

(10) Validate and verify the throughput, timing, memory usage, resource usage, and other resource issues of the RSC and other installed software components for the specific target environment.

(11) Keep the certification authority and designees (if applicable) informed of the project status and deviations from the approved plans. This coordination supports timely reviews by the certification authority and/or designees (if applicable) and approval of changed plans.

(12) At the end of the project, the applicant should submit the system level configuration index, SCI, SAS, and any other software life cycle data requested by the FAA. The SAS should include the information provided in Section 11.20 of RTCA/DO-178B for the system's software. The SCI and SAS should identify that the RSC has been included in their project, should identify the configuration of the RSC, and should identify the documentation configuration to support the RSC and other software components used in the system. Additionally, the SAS should include a description of how RTCA/DO-178B objectives that were not fully met by the developer of the RSC have been completely satisfied by the integrator or applicant.

c. Certification authority acceptance of the first version of a RSC. The certification authority or designee (if delegated) should follow these guidelines when accepting a RSC in its first installation.

(1) Work closely with the applicant, integrator, and RSC developer to ensure that they follow the guidelines outlined in sections 8.a and 8.b of this notice.

(2) Involve directorate personnel, headquarters personnel, technical specialists, and/or national resource specialists as needed, to address policy and technical issues in the project.

(3) Review the RSC developer's plans and the applicant's and/or integrator's plans to ensure that the objectives of RTCA/DO-178B will be satisfied when the plans are followed.

(4) Perform on-site or desk reviews of the software life cycle data and the involved organization's capability of the RSC developer, applicant, and integrator, as needed, to ensure compliance to the RTCA/DO-178B objectives.

(5) When the RSC developer, RSC integrator, and applicant satisfactorily complete their development and compliance activities, approve the applicant's, integrator's, and RSC developer's data, as in a typical software program, for the target system software.

(6) The certification authority should provide an acceptance letter for the RSC, addressed and submitted to the RSC developer. A copy should be provided to the project integrator and applicant. This letter should document the initial acceptance of the RSC and the suitability of use by other certification authorities, designees, applicants, and integrators who will address reuse of the RSC. The letter should include the information listed below:

(a) The RSC document numbers and revision levels approved (e.g., the SCI number and revision; the SAS number and revision; and any additional configuration information not included in the SCI), and a description of how the RSC was used for the target system.

(b) The RSC developer's name and contact information.

(c) The target system and project description, number, etc., for the initial approval.

(d) Assumptions made by the RSC developer during the acceptance, including a reference to the RSC developer's SAS, which should include assumptions for each applicable RTCA/DO-178B objective. The assumptions should be detailed enough that other certification authorities, RSC integrators, and applicants could apply the information to subsequent projects.

(e) Summary of technical or policy issues during the initial acceptance and how those were addressed.

(f) Summary of additional activities performed by the integrator and applicant to assure the RSC for the target system.

(g) Contact information for the certification authority who can address any future questions about the RSC acceptance and subsequent reuse.

(h) Software level of the RSC, any RSC limitations, and known safety or operational issues of the RSC.

d. Certification authority acceptance of a previously accepted RSC. The certification authority and the designee (if delegated) should follow these guidelines, when accepting a previously accepted RSC:

(1) Review the certification authority letter to the RSC developer documenting the initial acceptance. This letter may be obtained from the RSC developer or the certification authority that originally issued the acceptance.

(2) Contact the certification authority as documented in the letter to discuss project details and to address any questions, if needed.

(3) Work closely with the RSC applicant and integrator to ensure that they follow the guidelines outlined in section 8.b of this notice and to address any additional certification issues.

(4) Involve directorate personnel, headquarters personnel, technical specialists, and/or national resource specialists, as needed, to address policy and technical issues in the project (particularly for Level A and B projects).

(5) Review the integrator's and/or applicant's plans to ensure that when followed: (a) the objectives of RTCA/DO-178B will be satisfied, and (b) the assumptions and requirements documented for the RSC and for other software components used in the target system will be satisfied.

(6) Perform on-site and desk-top reviews of the integrator's and/or applicant's data and organizations capability, as needed, to ensure: (a) compliance to the RTCA/DO-178B objectives and approved plans, and (b) compliance with the assumptions and requirements documented for the RSC and other software components.

(7) When the applicant and integrator satisfactorily complete the integration and compliance activities, accept the applicant's and integrator's data, as in a typical software project, for the overall system software.

(8) Inform the certification authority who originally accepted the RSC of the subsequent software acceptance and report any issues that arose during the acceptance.

9. COMMON REUSE ISSUES AND CONSIDERATIONS. Throughout the acceptance of the RSC, a number of technical issues and questions will arise. This section provides some examples of common issues that should be addressed in the RSC and system-level PSACs. This section focuses on global reuse (i.e., issues that affect multiple RTCA/DO-178B objectives). This is not an exhaustive list, since each project will have its own specific issues.

a. High-level software requirements definition.

(1) A number of objectives in RTCA/DO-178B involve activities based on the high-level software requirements. For many projects there are multiple manufacturers involved. This can result in numerous requirements definitions with varying levels of abstractions. For example, a program could have the following hierarchical requirement levels: (1) National Airspace (e.g., satellites or ground stations) operational requirements; (2) Aircraft level requirements; (3) Line replaceable unit (LRU) requirements; (4) Software application requirements; (5) Reusable software component requirements; and (6) Software module requirements (e.g., lower-level software components, functions, and procedures). Figure 1 illustrates these multiple requirements levels.

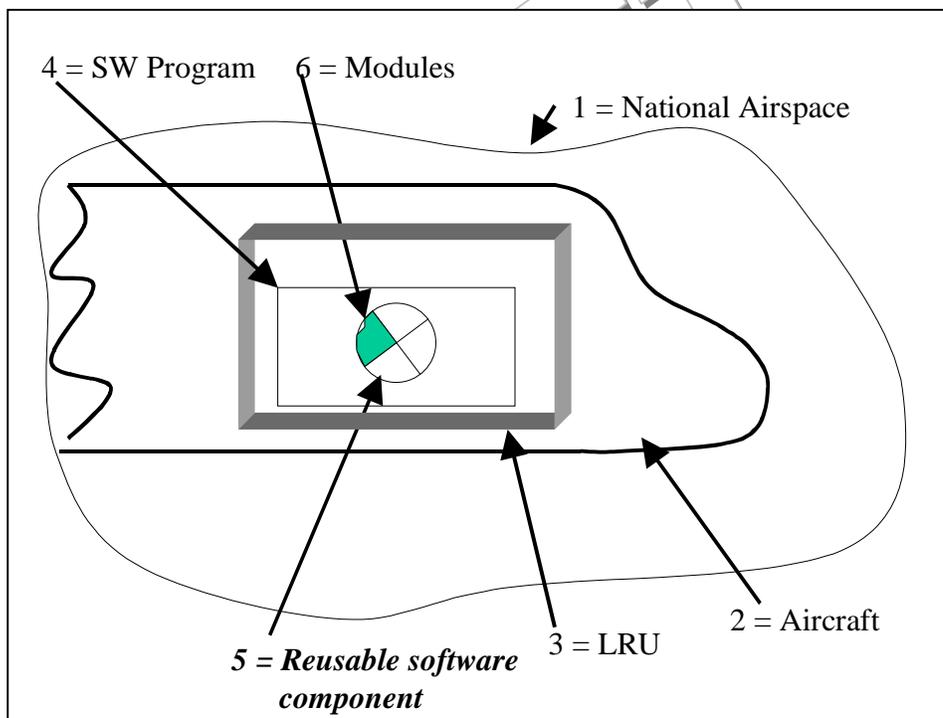


Figure 1. Example of Numerous Requirement Levels

(2) In the traditional airborne software world, levels 2 (usually captured in Aircraft System Requirements) and 3 (usually captured in LRU or system requirements) tend to be most readily recognized. RTCA/DO-178B was written with these levels in mind. In RTCA/DO-178B, “high-level requirements” are defined as “Software requirements developed from analysis of system requirements, safety-related requirements, and system architecture.” These system requirements

that are used as a starting point to develop the high-level requirements refer to the LRU and, in some cases, include aircraft-level requirements. They generally specify the functions for the software executing on a single processor and provide the basis for development of the software design, including the software architecture and low-level requirements. “Low-Level Requirements” are defined as, “Software requirements derived from high-level requirements, derived requirements, and design constraints from which source code can be directly implemented without further information.”

(3) However, in the case of the RSC, part of the software application executing on the processor was developed by a different organization or company than that developing the LRU to yet a different set of requirements. The question arises: “Of all the levels of software requirements refinement, which constitute the high-level requirements and who is responsible for verifying that they are properly implemented?”

(4) A number of RTCA/DO-178B objectives address “high-level requirements.” In order to meet the objectives, it is necessary to identify the appropriate refinement level for the requirements and their verification, and who will ensure they are met for the target system.

(5) In a traditional project, the requirements for a software application executing on a single processor within the LRU (sometimes identified as a Computer Software Configuration Item (CSCI)) would be the “high-level” software requirements allocated from the system requirements. For the integrator, this is still true. They will have the high-level software requirements that call out the RSC, as well as any other software components used for the target system. The LRU manufacturer is responsible for making sure these high-level software requirements are met for the LRU.

(6) The RSC developer develops the requirements for their software with the intent of it being used in many different LRUs. The RSC developer does not know how their software requirements are going to be integrated into the LRU requirements structure. The RSC requirements could be categorized at the LRU-level as high level requirements or low level requirements. This will vary between different LRU applications for the same RSC requirements.

(7) Derived requirements are defined in RTCA/DO-178B as “Additional requirements resulting from the software development processes, which may not be directly traceable to higher level requirements.” High-level requirements, by definition, should be traceable to system-level requirements. The RSC developer cannot identify from which system-level requirements their software requirements were derived. Therefore, for the RSC developer, their software requirements should be considered the derived high-level software requirements and should be verified as such. The RSC developer may also develop some derived low-level requirements. These should be identified to the users (integrators, applicants) of the RSC so that they can be included with all of the derived system requirements. All of the derived requirements, regardless of their source, should be addressed as indicated in RTCA/DO-178B.

(8) The LRU manufacturer is responsible for identifying which of the RSC derived high-level requirements can be correlated with the LRU high- and low-level requirements (system requirements). The remaining requirements will be considered derived. Once all of the requirements have been correlated and identified, they should be verified to ensure that the RSC properly interfaces with the other LRU software and hardware components.

(9) To summarize, the RSC developer should handle its software requirements as high-level requirements, lacking any agreements between them and the integrator or applicant. They should also identify any derived and low-level requirements. Then, it is the LRU manufacturer's responsibility to integrate the RSC requirements into the LRU requirements structure. Traceability between the RSC requirements and the LRU requirements structure should be documented. The integrator and applicant should satisfy all of the traceability and verification objectives of RTCA/DO-178B for the different levels of requirements refinement.

NOTE: The RSC developer might also consider producing system-level requirements and including them in their RSC package. The RSC system-level requirements could then be used efficiently by the integrator and applicant.

b. Re-verification issues. To determine the re-verification position, the following assumptions are made: source code is not changed, high-level requirements are not changed, low-level requirements are not changed, derived requirements are not changed, software architecture is not changed, and the applicant and integrator are using the same high-level requirements for the RSC as did the RSC developer. If these assumptions are true and a different compiler or a different target environment is used, two questions arise and are addressed below:

(1) How much re-verification is required with a different compiler type or optimization or a different target environment and why?

(a) In general, applicable verification done by review, analysis, and test should be repeated when a different target computer, a different target environment, and/or a different compiler type or optimization is used. However, for a given project, it might be possible to use analysis to show that specific verification objectives and data will not be affected by a compiler or target environment change. Analysis situations should be handled on a project-by-project basis.

(b) Also in general, verification activities independent of the target computer or environment, such as reviewing objectives and data for "Source Code conforms to standards" (Table A-5, objective 4, 6.3.4d.), need not be repeated when the software is recompiled and installed in a new target computer and/or environment. There are certain exceptions: it should be ensured that the new target environment system requirements are compatible with the software requirements of the RSC. If they are not completely compatible, modifications may be needed to the RSC; refer to Section 10 of this notice.

(c) It should be noted that some typical issues when addressing re-verification are: (1) run-time and compiler libraries (i.e., are they the same as those used in the RSC development?);

and (2) resource issues (e.g., are there non-deterministic, dynamic memory allocation algorithms with the RSC that could create resource (e.g., memory, execution time) issues in the new target environment?).

(d) Re-verification issues should be addressed in the PSAC, verification plan, and any other necessary plans.

(2) Will structural coverage analysis need to be repeated on the target computer?

(a) **Level A software.** For Level A software, structural coverage is typically performed on the source code, using requirements based tests (RBTs) to exercise the code. An additional analysis is performed to demonstrate source to object code traceability per RTCA/DO-178B Section 6.4.4.2. If the RBTs are executed without change and with correct results, the structural coverage analysis performed on the source code should not need to be repeated since the source code did not change. However, source to object code traceability should be re-evaluated for the different target, environment, and/or compiler type or optimization.

(b) **Level B and C software.** If the RBTs are executed without change and with correct results, structural coverage analysis should not need to be repeated for Levels B and C software, since the source code has not changed. However if the RSC will be interfacing with other software components or other hardware devices in the new application, data coupling and control coupling analysis may need to be repeated for those new or modified interfaces.

c. **Interface documents.** The RSC developer should provide interface documents. These should explicitly define what activities are required by the integrator and/or applicant to ensure that the RSC will function in accordance with its requirements. Typical items included in interface documents are any configuration parameters, any restrictions on tools, additional verification needed, memory and timing requirements, and any other external resources required by the RSC for proper functioning and performance. Additionally, the interface documents will define the communication mechanisms between the RSC and other software programs and the communication protocols with hardware components. Typical items included are accessible variables and their characteristics; variables and data required from the system and their characteristics (inputs to RSC); bus and input/output ports and devices; access mechanisms; etc.

d. **Partitioning and protection considerations.** Although partitioning and protection will most likely be a function at the system level, the RSC may need some partitioning and protection. For example, there may be some maintenance code that is at a different software level than the operational flight program for the RSC. In some cases, the RSC might have specific protocols that facilitate protection and partitioning. These will need to be documented and evaluated by the integrator, applicant, and certification authorities.

e. **Safety issues.** Safety assumptions should be well documented by the RSC developer in the PSAC and SAS. The RSC developer should also produce a safety assessment based on how they intend the RSC to be used to document any known issues that could affect safety.

f. Coupling and Cohesion. Data and control coupling and cohesion issues between the RSC and other integrated software and hardware components should be carefully addressed. For example, modification to internal RSC data and structures should be prohibited or tightly controlled.

g. Use of qualified tools. If qualified tools are used for the development and/or verification of the reusable software component (RSC), the reuse of supporting tool qualification data should be considered during the acceptance of the RSC. RTCA/DO-178B Section 12.2 and FAA Notice 8110.91 provide additional information on the tool qualification process and the supporting documentation.

(1) When qualified tools are used for the development and/or verification of a RSC, the Tool Qualification Plan and the Tool Accomplishment Summary (or PSAC/SAS for verification tools) should document any portions of the tool qualification that are to be completed by the applicant. For example, test procedures and cases might have some target dependencies and additional verification effort by the integrator/applicant should be performed.

Note: Some developers have found that packaging the qualification data for each tool helps with reuse. For example, each verification tool used with a RSC might have its own Tool Qualification Plan, Tool Operational Requirements, and Tool Accomplishment Summary.

(2) The following tool qualification data should be provided to the applicant for all tools used with the RSC to be qualified:

- (a) All tool plans,
- (b) Tool Operational Requirements, and
- (c) Tool Accomplishment Summary.

(3) All other tool data should be available to the applicant and certification authority, as needed, to support continued airworthiness.

h. Deactivated code. Any information about deactivated code and the associated deactivation mechanisms should be identified. Since the RSC may have many features to satisfy a broad audience, a mechanism is needed to tailor the RSC to the specified requirements of an applicant's application. This could result in sections of deactivated code which will need to be addressed as part of the overall software approval process.

10. CHANGES TO REUSABLE SOFTWARE COMPONENTS.

- a. RSCs will likely change at some point in time.

b. When a RSC is changed, the original reuse status is no longer applicable to the changed component.

c. When a RSC is changed, it should be accepted as a part of a project, following the guidelines of this document, assuming the involved parties want to continue to keep reuse status on their changed component.

d. When a RSC is changed, a change impact analysis should be performed to identify the changed and affected aspects of the software. Notice 8110.85 provides information regarding the types of activities involved in performing a change impact analysis.

e. If the applicant or integrator decides to make changes to the RSC without the RSC developer, they become responsible for the entire component.

f. Changes to a RSC for one project may or may not affect other projects using the RSC. Change to RSCs as a result of airworthiness directives (ADs) should be coordinated with the RSC developers, users of the RSC, and the appropriate certification authorities' offices to determine how the AD applies to other projects.

11. CONCLUSION. The information and procedures described in this notice promote clarification and consistent application of AC 20-115B for the acceptance of reusable software components in airborne systems and equipment. This notice does not replace or supersede AC 20-115B or RTCA/DO-178B.

David W. Hempe
Manager, Aircraft Engineering Division,
Aircraft Certification Service

APPENDIX 1 - Example Format for RSC Developer's Table

<i>178B Obj #</i>	<i>Obj Description</i>	<i>Credit Sought</i>	<i>Assumption</i>	<i>Means of Compliance for the Objective</i>	<i>Activities Remaining For Integrator/Applicant</i>
1-1	Software development and integral processes activities are defined.	NOTE 1	NOTE 2	NOTE 3	NOTE 4
1-2	Transition criteria, inter-relationships and sequencing among processes are defined.				
1-3	Software life cycle environment is defined.				
1-4	Additional considerations are addressed.				
1-5	Software development standards are defined.				
1-6	Software plans comply with this document.				
1-7	Software plans are coordinated.				
2-1	High-level requirements are developed.				
ETC.					

NOTE 1: Include if FULL, PARTIAL, or NO credit is being sought for the RSC. Reference section 8.a.(1)(a) of this notice.

NOTE 2: List all assumptions made for the credit claim. Reference section 8.a.(1)(b) of this notice.

NOTE 3: List data that documents the compliance to this objective. Reference section 8.a.(1)(c) of this notice.

NOTE 4: List the activities remaining for the integrator and/or applicant to complete the objective. This should be in enough detail that the integrator and/or applicant and the certification authority can clearly understand what remains for the overall acceptance of the system using the RSC. Reference section 8.a.(1)(d) of this notice.

DRAFT