U.S. Department
of Transportation
**Federal Aviation
Administration**

Advisory Circular

AC 20-148

REUSABLE SOFTWARE COMPONENTS

December 7, 2004

CONTENTS

1. PURPOSE.

a. This advisory circular (AC) provides one acceptable means of compliance, but not the only means, for reusable software component (RSC) developers, integrators, and applicants to gain:

- The Federal Aviation Administration's (FAA) "acceptance" of a software component that may be only a part of an airborne system's software applications and intended functions.

- Credit for the reuse of a software component in follow-on systems and certification projects, including "full credit" or "partial credit" for compliance to the objectives of RTCA/DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*.

b. Like all advisory material, this AC is not mandatory and does not constitute a regulation. Because the means of compliance in this AC is not mandatory, the term "must" applies only to the applicants, integrators, and RSC developers who choose to follow the method in this AC.

2. MOTIVATION FOR THIS GUIDANCE.

a. Because of economic incentives and advances in software component technology, software developers want to develop an RSC that can be integrated into many systems' target computers and environments with other system software applications, as determined by the integrator or applicant. In these cases, an RSC developer may partially satisfy the applicable RTCA/DO-178B objectives, while the integrator or applicant completes and shows the compliance for the integrated software package, systems aspects, and aircraft certification. Examples of potential RSCs include software libraries, operating systems, and communication protocols.

b. The guidance in this AC ensures that systems using RSCs meet all applicable RTCA/DO-178B objectives.

> **NOTE:** The reuse concept in this AC may apply to verification and development tools. Applicants and tool developers must discuss with the FAA the details of each reusable tool qualification project. Because tools differ from airborne software, there are other concerns to address when trying to reuse tool qualification data. The FAA plans to address tool reuse in future guidance.

3. DOCUMENT OVERVIEW.

a. Paragraphs 1 through 3 explain the purpose of this AC, discuss the motivation for the guidance, and provide a document overview.

b. Paragraph 4 provides general guidelines for receiving FAA acceptance of an RSC.

c. Paragraphs 5 through 7 guide the RSC developers, integrators, and applicants on developing or using an RSC.

d. Paragraphs 8 through 10 provide typical activities the RSC developers, integrators, and applicants can expect from the FAA for the first acceptance of an RSC and its subsequent use.

e. Paragraph 11 discusses common issues to address when developing and using RSCs. These issues may affect multiple RTCA/DO-178B objectives. Paragraph 11 is not an exhaustive list of issues that may arise, since each project will have its own specific issues.

f. Paragraph 12 addresses changes to an RSC.

g. Paragraph 13 considers concurrent uses of an RSC.

h. Paragraph 14 lists FAA and industry documents that guide how to show an RSC complies with regulations for software components.

i. Appendix 1 defines the terms used in this AC. **Please review this appendix before reading the AC.**

j. Appendix 2 lists the acronyms used in this AC.

k. Appendix 3 provides a sample format for an RSC developer to document RTCA/DO-178B credit.

4. GENERAL GUIDELINES FOR RECEIVING FAA ACCEPTANCE OF AN RSC.

a. The FAA may grant acceptance for an RSC after: (i) all stakeholders comply with this AC and (ii) the FAA (or authorized designee) does not identify any installation, safety, operational, functional, or performance concerns. The term "stakeholders" means the applicant, integrator, RSC developer, and certification authority. To show acceptance for an RSC, the FAA will issue an RSC acceptance letter. The FAA will not write the letter until it grants a certification or authorization for a product or equipment using the RSC. If the RSC is unchanged and meets the limitations in the RSC acceptance letter, it may be reused without additional FAA review of the RSC data. This can only happen when no safety, installation, operational, functional, or performance concerns are identified in the subsequent uses.

b. This AC requires that an RSC being considered for acceptance has its own set of software life cycle data. This AC applies to the "acceptance" of an RSC to support the approval of airborne systems and equipment. Since an RSC is integrated with other systems' software, the RSC must comply with the applicable regulations, guidance, and RTCA/DO-178B objectives.

c. The FAA will grant acceptance for an RSC as part of the approval for type certificates (TC), supplemental type certificates (STC), amended supplemental type certificates (ASTC), amended type certificates (ATC), and technical standard order (TSO) authorizations. For TSO authorized articles that are highly integrated into the aircraft or that have aircraft dependencies, the FAA typically will not grant the RSC acceptance letter until it grants installation approval to the TSO authorized article and the RSC. This is necessary because of the aircraft system's

complexity and aircraft-specific dependencies on installed systems and equipment. The FAA grants installation approval for a TSO authorized article as part of a TC, ATC, STC, or ASTC.

d. The first acceptance of an RSC must be performed during a project (such as a TC, ATC, STC, ASTC, or TSO authorization project). Stakeholders may need to provide extra resources. Later acceptance of the RSC for a different system or project will likely require less effort and resources, if stakeholders follow the guidance in this AC.

e. This reuse guidance applies only when all stakeholders agree that the software component is reusable. The RSC Plan for Software Aspects of Certification (PSAC) and the first applicant's system-level PSAC must document that stakeholders agree on the proposed means of showing the RSC compliance with this AC. Both plans also help define the communication channels and roles among stakeholders. Agreeing on the reuse is important because the first applicant will likely use more resources to qualify the component as reusable. If there is no agreement, then stakeholders should follow the traditional approach to software development and approval for all software in the system (see paragraph 5 of this AC).

f. Each RSC developer's project will have different limitations, needs, and issues. For example, one developer may package the software life cycle data so the data fully satisfy a particular objective of RTCA/DO-178B. Another RSC developer may only partially satisfy that same objective. This may be due to some project-specific issues, or more coordination with the integrator to add to the RSC developer's efforts. Paragraphs 5 through 7 of this AC guide the RSC developer, integrator, and applicant. The guidelines are flexible enough to satisfy the multiple needs of the RSC developer, integrator, and applicant. However, the guidelines are also detailed enough to ensure that the RSC developer, integrator, and applicant address relevant certification, compliance, and safety issues.

g. Applicants must submit compliance data, coordinate and communicate with the FAA, and serve as the certification liaison for the project. However, the certification authority and the RSC developer (with the applicant's involvement) may need to communicate about the reuse aspects of the project.

h. Acceptance of an RSC for one project does not guarantee acceptance on a later project. Applicants must consider installation, safety, operational, functional, and performance issues for each project. If concerns arise in any of these areas, the FAA may need to reassess RSC life cycle data. Also, applicants must address compliance to all applicable RTCA/DO-178B objectives, guidance, and regulations relevant to their project.

i. Although the FAA coordinated this AC with international certification authorities, international programs or approvals may require activities not addressed in this AC. Applicants should closely coordinate international projects with all applicable certification authorities.

j. The integrator and applicant should be aware that:

(1) An RSC is unlikely to satisfy all objectives of RTCA/DO-178B.

(2) The integrator or applicant may need to validate the RSC developer claims.

(3) The integrator or applicant may need to provide additional resources to show compliance to RTCA/DO-178B objectives for systems containing RSCs.

(4) The communication paths and division of responsibilities can be complex, when using an RSC.

(5) Other regulations, guidance, and agreements – beyond RTCA/DO-178B – may apply to their system and its aircraft installation approval. These may depend on the date of application for the certification; the system they are proposing; the introduction of novel design, technology, or methods; or other factors. The applicant must show that all components of their system, including RSCs, comply with all applicable regulations.

k. The FAA recommends that the RSC developer, integrator, and applicant use RTCA/DO-178B as the means of compliance for the software aspects of the initial approval or future certification approvals of systems containing RSCs. As described in paragraph 14b of this AC, if an alternate means is proposed, this AC may not apply.

5. GUIDELINES FOR THE RSC DEVELOPER. Traditionally, RSC developers provided one of two ways to show that their RSC complied with RTCA/DO-178B:

- By resubmitting the RSC software life cycle data and repeating the work for each system's application; or
- By providing traceability through the TC, ATC, STC, ASTC, or TSO approval back to the desired data, and justifying the validity of their processes and data from the original approval basis to the new approval basis for each system.

Before issuance of this AC, there were no procedures for RSC developers to transfer their accepted data directly from one project to the next and across company boundaries. This AC addresses the reuse of software components and software life cycle data across company or division boundaries. The RSC developer must do the following:

a. Produce a PSAC for the RSC as early as possible in the project. The RSC PSAC must:

(1) Include the information in Section 11.1 of RTCA/DO-178B;

(2) Detail the RSC developer's plans for satisfying each applicable RTCA/DO-178B objective;

(3) Identify which objectives the RSC developer will not satisfy, and which objectives they will partially satisfy;

(4) Clearly state the RSC developer's agreement to develop the RSC for reuse in future projects;

(5) State the intent to comply with this AC;

(6) Define the failure conditions, safety features, protection mechanisms, architecture, limitations, software levels, interface specifications, and intended use of the RSC; and

(7) Describe the proposed certification liaison process (including communication and coordination focal points) to all involved stakeholders.

b. Consider and address, as applicable, the common reuse issues in paragraph 11 of this AC.

c. For each RTCA/DO-178B objective applicable to the software level, document in the RSC PSAC the information in paragraphs 5d(1) through 5d(4). Include enough detail to obtain certification authority concurrence and to enable the integrator or applicant to use the RSC. The RSC developer may include this information in a table, with columns for the following (see a sample format in appendix 3):

- RTCA/DO-178B objective reference;
- RTCA/DO-178B objective description;
- Amount of credit being sought (full, partial, or no credit);
- Assumptions;
- Means of compliance; and
- Remaining activities the integrator or applicant must complete.

d. List preliminary information in the RSC PSAC, since resource-specific, target computer-specific, and system-specific issues may be uncertain early in the project. Update this information in RSC PSAC revisions and the RSC Software Accomplishment Summary (SAS), when the RSC is completed. Some reuse details may not be finalized until the end of the project. RSC developers must thoroughly document the following information for each applicable RTCA/DO-178B objective for review by certification authorities (and authorized designees) and for use by integrators or applicants:

(1) Credit Being Sought for the Objective. The RSC PSAC or referenced document must specify if the RSC developer is seeking full, partial, or no credit for the objective. Full credit means being able to satisfy an objective completely using the RSC data package, and to show that all associated assumptions are valid. If integrators need to do more, then they cannot claim full credit. This is true even when the RSC developer fully specifies what integrators must do. Also, if integrators or applicants do not satisfy the assumptions, they cannot get credit.

(2) RSC Developer's Assumptions on How the Integrators or Applicants Will Use the RSC. The RSC developer must provide enough justification to ensure that the original acceptance is valid if the assumptions are satisfied. For example, the RSC developer may assume the source code, compiler type, and compiler options will remain the same. However, if integrators or applicants do not meet these assumptions, they cannot get credit for the applicable RTCA/DO-178B objective.

(3) Means of Compliance for the Objective. The RSC PSAC and SAS must document what software life cycle data support compliance for each applicable objective. As

evidence of compliance, the RSC PSAC and SAS must provide document titles, version numbers, and/or a description of the data.

(4) Activities Remaining for the Integrator or Applicant. The RSC PSAC and SAS must document what an applicant or integrator must do to satisfy any partial or unsatisfied objectives fully.

e. Document the following safety-related items in the RSC PSAC and SAS:

(1) The software levels for the RSC.

(2) An analysis of all interfaces and settable parameters. The analysis should describe the functional and performance effects of these parameters and interfaces on the user. The analysis should document required actions by the user to ensure proper operation.

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

(4) Any safety, operational, functional, or performance assumptions that support the use of the RSC (see paragraph 5f).

(5) Any new or novel concepts, methods, and technologies for developing the RSC.

f. Produce an analysis of the RSC's behavior that could adversely affect the users' implementation (for example, vulnerabilities, partitioning requirements, hardware failure effects, requirements for redundancy, data latency, and design constraints for correct RSC operation). The analysis may support the integrator's or applicant's safety analysis.

g. Get agreement (as early as possible) from all stakeholders for the first application. Do this by coordinating the following with the certification authority, designees (if delegated), and the applicant or integrator:

(1) The RSC PSAC;

(2) Any other RSC plans (for example, Software Development Plan (SDP), Software Verification Plan (SVP), Software Quality Assurance Plan (SQAP), and Software Configuration Management Plan (SCMP)); and

(3) Software development standards (that is, requirements, design, and coding standards).

h. Develop the RSC per the approved plans. As previously stated, the RSC developer must produce the RTCA/DO-178B software life cycle data and documentation in paragraph 6 of this AC (such as plans, standards, development data, verification data, quality assurance records, and configuration management records).

i. Inform the certification authority, designees (if delegated), integrator, and applicant of development progress and any deviations from plans, to allow for timely reviews and adjustments as necessary.

j. Submit the RSC Software Configuration Index (SCI) and the RSC SAS to the certification authority through the project applicant, when completed. The RSC SAS must include or refer to the software life cycle data in RTCA/DO-178B, Section 11. Additionally, the RSC SAS must show how the RSC complies with this AC.

6. DATA THE RSC DEVELOPER MUST SUPPLY TO THE RSC INTEGRATOR OR APPLICANT. The RSC developer must supply the appropriate software life cycle data to the integrator or applicant. These data will support acceptance of the RSC in the context of the software aspects of certification of the airborne systems using the RSC. RSC developers must supply the data in paragraphs 6a through 6i to the RSC integrator or applicant, and to the certification authority on request, unless they use a data/software escrow approach per paragraph 6j(2) of this AC:

a. The type design data in Section 9.4 of RTCA/DO-178B for the RSC (that is, software requirements data, design description, source code, executable object code, SCI, and SAS).

b. The RSC PSAC, which identifies the credit sought for each RTCA/DO-178B objective.

c. Interface description data (for example, interface control document and porting guide). The interface description data include any hardware and software resource requirements (such as timing and memory) and applicable analyses, verification procedures, and verification cases.

d. Installation or integration procedures and limitations, to ensure the RSC will be properly used, integrated, and installed. They must be detailed enough to identify unique aspects of the installation or integration. The limitations and procedures must include, as a minimum:

(1) Equipment specifications for proper operation and performance of the RSC, including verification activities the integrator or applicant must perform to ensure the equipment meets specifications;

(2) A list of any RSC subcomponents, as defined by RTCA/DO-178B, Section 11.16; and

(3) Instructions for periodic maintenance and/or calibration needed for continued airworthiness after installing the software on the target environment.

e. Data to support the integrator's or applicant's completion of partially satisfied or unsatisfied objectives. As an example, if seeking partial credit for objective 1 of RTCA/DO-178B, Table A-1, the RSC developer clearly defines to the integrator or applicant what that partial credit entails. Also, the RSC developer defines what the integrator or applicant needs to do to gain full credit for the installation. The data supporting any "full" or "partial" credit must also be available to the integrator or applicant.

f. Software verification results, verification cases, and verification procedures, especially for verification activities that the integrator or applicant must repeat for the integrated software installed on the target computer environment. Examples of verification activities the integrator or applicant carries out include:

- Data coupling analysis;
- Control coupling analysis;
- Timing analysis;
- Memory analysis;
- Software integration testing;
- Hardware-software integration testing; and
- Robustness testing of RSC functions, including safety and protection features.

g. The verification data should include a list of test cases and procedures affected by any settable parameters. The integrator or applicant should consider the total requirements for system and subsystem testing. Within this context the integrator or applicant should address:

(1) Applicable credit for reusable tests of the RSC.

(2) Retests where new settings or parameters may affect the requirements, code, function, performance, or protection features.

(3) Analyses of data coupling and control coupling of the RSC, including guidance for the integrator or applicant. This guidance helps with the data coupling analysis and control coupling analysis of the RSC integrated with other airborne software components of the system.

(4) Development of new test cases and procedures to complete all tests and test coverage objectives, including guidance for the integrator or applicant. The guidance will help show normal range and robustness testing and test coverage objectives for the entire integrated airborne software.

h. Open problem reports on the RSC and analysis of any potential functional, operational, performance, and safety effects. The RSC developer should document this information in the RSC SAS, and – if the developer knows this information at the beginning of the project – include it in the RSC PSAC.

i. The RSC developer must develop a data sheet for the RSC and submit it to the FAA. The integrator or applicant may attach the data sheet to the FAA acceptance letter. This data sheet must concisely summarize:

- RSC functions;
- Limitations;
- Analysis of potential interface safety concerns;
- Assumptions;
- Configuration;
- Supporting data;

- Open problem reports;
- Software characteristics; and
- Other relevant information that supports the integrator's or applicant's use of the RSC.

j. RSC developers must also address the following data-related items (although the items may not result in submittals):

(1) Any RTCA/DO-178B software life cycle data not listed above – but used for software development and approval – must be available to the applicant, integrator, and certification authority (for example, Software Quality Assurance and Software Configuration Management records).

(2) Regardless 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 must be available for certification authority (and authorized designee) review and inspection. The RSC developer may set up a process (such as a data or software escrow) for making some data (such as design description or source code) available to the applicant, without supplying that data to the applicant. All stakeholders must agree on the process of escrowing a subset of the data in paragraphs 6a through 6h, and document the process in the RSC PSAC and system-level PSAC. All data must be accessible to the certification authority (and authorized designees) to determine compliance or any safety or operational problems with the target system (see 14 CFR § 21.277). The data may also need to be available to the applicant, if the target system or RSC requires modification (see 14 CFR § 21.301 through § 21.305, and FAA Order 8110.4).

(3) The RSC developer must identify and maintain data to support changes to the RSC. For example, if the developer goes out of business, these data will support continued airworthiness and operational safety. Title 14 CFR part 21, Certification Procedures for Products and Parts (as supplemented by FAA Order 8110.4, Type Certification (Chapters 2 and 3)), discusses how to issue and preserve type design data for maintaining the continued airworthiness of aircraft products.

(4) The RSC developer must retain and maintain a list of all integrators and applicants buying or using their components to support continued airworthiness across multiple products. The RSC developer, integrator, and applicant must set up a process to share in-service problem reports that support operators required to comply with 14 CFR § 21.3, and that support paragraphs 7q through 7s of this AC. The RSC developer and users must develop an agreement to support continued airworthiness of the systems using the RSC.

7. GUIDELINES FOR THE INTEGRATOR AND APPLICANT USING THE RSC.

Sometimes the integrator and applicant are the same company or organization, and sometimes they are separate entities. Below are the guidelines for the integrator and applicant. The applicant ensures these items are completed, even if an integrator performs some tasks. The applicant or integrator must perform the following for each RSC integrated into their system application:

a. Integrate the RSC developer's plans, data, limitations, compliance statement, mapping to RTCA/DO-178B objectives, software approval approach, and other relevant information (such as RSC acceptance letter and data sheet) into their own software life cycle data.

b. Specify the software life cycle data that RSC developers must provide to support their project and continued airworthiness. See a list of these data in paragraph 6 of this AC.

c. Produce a system-level PSAC (or equivalent certification plan) for the target system, including the information in RTCA/DO-178B, Section 11.1. The system-level PSAC must include the integrator's or applicant's plans to address compliance with all RTCA/DO-178B objectives, regulations, and guidance for the RSC and other software components of the target system. Also, the system-level PSAC must clearly state the agreement that the RSC was developed with the intent to be reusable in other projects and that the applicant or integrator intends to comply with this AC.

d. Produce other system-level software plans (such as SDP, SCMP, SVP, and SQAP) for their target system. Each plan must address the RSC integration and other software components used. For example, the system-level SVP must cover the overall software verification program, plus any verification required to integrate the RSC and other components, and the credit proposed for the RSC developer's verification.

e. Evaluate the safety, operational, performance, and functional impacts of the issues in the RSC developer's PSAC, SAS, and safety analysis data; and determine the applicability and severity of these impacts on the specific application and system.

f. Determine any other impacts for the specific application.

g. Propose risk mitigation, system architectural design features, protection mechanisms, and other assurance methods to address those risks.

h. Address all safety, operational, functional, and performance issues during the development of the system.

i. Coordinate all plans and standards (as needed) with the certification authority and designees (if delegated) to get agreement on the project.

j. Follow the approved plans and standards. Coordinate any deviations from the plans or standards with the certification authority (and authorized designees) before implementation.

k. Analyze open problem reports on the RSC (including development problem reports and in-service problem reports), other software components, hardware, and the system. Ensure there are no safety, operational, functional, or performance effects from the RSC or other components in the specific application and system.

l. Validate that the assumptions the RSC developer made in the RSC SAS for RTCA/DO-178B objective credit are met. Show how the credit applies to the integrated system that uses the RSC, and complete the RTCA/DO-178B objectives identified as "partial" or "no"

credit in the RSC SAS. The applicant must ensure compliance to all applicable RTCA/DO-178B objectives for the integrated RSC.

m. Evaluate and address the common reuse issues in paragraph 11 of this AC for each particular application. Additionally, address any project-specific reuse issues.

n. Validate and verify the throughput, timing, memory usage, resource usage, and other resource items of the RSC and other installed software components for the specific target environment.

o. Inform the certification authority and designees (if applicable) of the project status and approved plan deviations. This communication supports timely reviews by the certification authority or designees (if applicable) and approval of changed plans.

p. Submit all SCIs, SASs, and other required software life cycle data to the certification authority (that is, submit both system-level and RSC data). The system-level SAS must include the information in Section 11.20 of RTCA/DO-178B for the system's software. Also, the system-level SAS must include a description of how the integrator or applicant for the entire integrated system completely satisfied RTCA/DO-178B objectives that the RSC developer did not fully meet. The system-level SCI and SAS must identify:

- The RSC is in the applicant's project;
- The configuration (including part and version numbers) of the RSC;
- The configuration (including part and version numbers) of other software components; and
- The software life cycle data configuration to support the RSC and other software components in the system.

q. Report in-service problems with the RSC to the RSC developer and the certification authority who granted the acceptance letter.

r. For subsequent use of the RSC, investigate the in-service experience related to the RSC to ensure there have been no safety-related problems connected with the RSC. For this purpose (see paragraph 6j(4)), evaluate relevant information, such as problem reports available to the RSC developer. The RSC developer and the certification authorities must know about all safety-related in-service experiences involving the RSC.

s. Establish a legal agreement with the RSC developer about continued airworthiness support, data ownership, and so on, to meet the regulations.

8. EXPECTATIONS FROM CERTIFICATION AUTHORITIES ON THE FIRST USE OF AN RSC. The RSC developer, integrator, and applicant should work closely with the certification authority throughout the RSC development and integration. To gain acceptance of an RSC in its first system installation, the certification authority will typically:

a. Coordinate and work closely with the applicant, integrator, and RSC developer to ensure they comply with this AC and other applicable regulations or guidance.

b. Involve directorate personnel, headquarters personnel, technical specialists, and chief scientific and technical advisors (CSTA), as needed, to address policy and technical issues in the project.

c. Review the RSC developer's, applicant's, and integrator's plans to ensure the applicable RTCA/DO-178B objectives, regulations, and guidance will be satisfied.

d. Perform on-site or desk reviews of the software life cycle data and the ability of the RSC developer, applicant, and integrator, as needed, to ensure compliance to the applicable RTCA/DO-178B objectives, guidance, and regulations.

e. Ensure the applicant and RSC developer establish a process to address any continued airworthiness and in-service problems (see paragraphs 6j(4), 7q, 7r, 7s, and 10h of this AC).

f. Approve data from the applicant, integrator, and RSC developer (as in a typical software program) for the system software, when the stakeholders satisfactorily complete their development and compliance activities. The approval will include approval of the data and software escrow processes, where applicable.

9. THE RSC ACCEPTANCE LETTER. If this AC is followed, upon successful certification of the product or authorization of the equipment using the RSC, the FAA will write an acceptance letter for the RSC and will submit it to the RSC developer. The certification authority will provide a copy to the applicant or integrator. This letter documents the initial acceptance of the RSC and its suitability for applicants or integrators to use in other certification projects. The acceptance letter typically includes:

a. The RSC document numbers and revision levels approved, and a general description of the RSC functionality and target environment. Examples of RSC document numbers and revision levels include the SCI number and revision; the SAS number and revision; and any configuration information not included in the SCI.

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

c. The name and contact information of the original RSC applicant or integrator, the airborne system and environment, and other relevant information about the initial acceptance of the RSC.

d. Assumptions the RSC developer made during the acceptance – typically done by including a reference to the RSC developer's SAS (or other appropriate data). The referenced document must include assumptions for each applicable RTCA/DO-178B objective. The assumptions must be detailed enough that other certification authorities, RSC integrators, and applicants could apply the information to later projects.

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

f. Summary of extra activities that the integrator and applicant performed to ensure the RSC meets requirements for the initial approval, including system bench and aircraft testing.

g. Contact information for the certification office that will address questions about the RSC acceptance and subsequent reuse.

h. Software level of the RSC, any RSC limitations, and known installation, safety, operational, functional, or performance issues of the RSC.

i. RSC data sheet, as described in paragraph 6i of this AC. The certification authority may attach a copy of the RSC data sheet to the acceptance letter.

j. Emphasis that acceptance of the RSC in one project is not approval in any other project. Subsequent users of the RSC must evaluate installation, safety, operational, functional, and performance aspects of the RSC in their application. Also, subsequent users of the RSC must evaluate complete compliance to all applicable RTCA/DO-178B objectives, regulations, and guidance for the RSC and other components in their system.

> **NOTE:** Certification authorities may encourage RSC developers to document some or all of the information in paragraphs 9a through 9j in their data sheet or RSC SAS. If RSC developers use this approach, then the certification authority can simply refer to the data sheet or RSC SAS (number, title, and revision number) in the acceptance letter. In either case, the RSC developer must supply the data sheet and RSC SAS to subsequent RSC users (applicants or integrators) per paragraph 6 of this AC.

10. EXPECTATIONS FROM THE CERTIFICATION AUTHORITIES ON THE SUBSEQUENT USE OF AN ACCEPTED RSC. When a previously accepted RSC is submitted for subsequent reuse in another product or project or by another integrator or applicant, the certification authority or the designee (if delegated) will:

a. Review the RSC acceptance letter that documents the initial acceptance. Get this letter from the RSC developer or the certification authority office that originally issued the acceptance.

b. Contact the certification office that performed the first acceptance (as documented in the acceptance letter) to discuss project details and to address any questions or concerns.

c. Coordinate and work closely with applicants or integrators to ensure they follow this AC's guidelines, address the common reuse issues in paragraph 11 of this AC, and address any other certification issues.

d. Ensure that installation, safety, operational, functional, and performance aspects of the RSC in the specific system have been analyzed and addressed.

e. Involve FAA Directorate personnel, headquarters personnel, technical specialists, or CSTAs, as needed, to address policy and technical issues in the project.

f. Review the integrator's or applicant's plans to ensure:

(1) The objectives of RTCA/DO-178B will be satisfied;

(2) Other applicable regulations, guidance, and agreements will be satisfied; and

(3) The assumptions and requirements documented for the RSC and for other software components used in the target system will be satisfied.

g. Perform on-site and desk reviews of the integrator's or applicant's data and organization's ability, as needed. This ensures compliance to the applicable RTCA/DO-178B objectives, regulations, guidance, and approved plans. It also ensures compliance with the assumptions and requirements documented for the RSC and other software components.

h. Evaluate the in-service experience related to the RSC, as described in paragraphs 7q through 7s. Address safety-related in-service experience and continued airworthiness concerns before accepting the new use of the RSC.

i. Accept the applicant's or integrator's data for the overall system software after they satisfactorily complete the integration and compliance activities.

j. Inform the original certification authority of later software acceptance, and report any issues that arose during the acceptance.

11. COMMON SOFTWARE REUSE ISSUES. Several issues may affect the reuse of software components. Below are some common issues; this is not an exhaustive list:

a. Requirements definition.

(1) RTCA/DO-178B discusses several types of requirements, including system requirements, safety-related requirements, high-level requirements, low-level requirements, and derived requirements. The RTCA/DO-178B discussion and objectives about requirements were developed with a traditional federated system in mind. In the traditional case, a single manufacturer typically develops and integrates the software. When RSCs and multiple stakeholders become involved in the software assurance process, determining the types and levels of requirements may become more difficult. Therefore, satisfying the RTCA/DO-178B objectives related to requirements requires special attention.

(2) Each RSC developer must establish a plan to satisfy the RTCA/DO-178B objectives related to system, high-level, low-level, and derived requirements. RSC developers will most likely not be able to satisfy RTCA/DO-178B objectives related to traceability and to compliance and consistency with system requirements. They also will most likely not be able to validate RSC derived requirements for the system's safety assessment process to ensure design decisions do not affect the system safety. This will likely result in more effort for the integrator or applicant.

(3) The RSC developer must clearly document in the RSC PSAC the means of addressing requirements. The integrator or applicant must adhere to those means. In the system-level PSAC, the integrator or applicant must clearly document the means of addressing requirements and the system safety assessment. The applicant should also coordinate both PSACs with the appropriate certification authorities as early in the program as possible.

b. Reverification.

(1) When an RSC is reused, the question of how much reverification to perform often arises. Reverification activities depend on the situation (such as same or different processor, same or different compiler, same or different compiler options, and so on). RSC developers should document their overall verification (and reverification) plans in the RSC PSAC. The RSC SVP should provide more details. The RSC PSAC and SVP should have enough detail for the certification authority to determine the approach will address the RTCA/DO-178B verification objectives. The integrator or applicant will also need to address verification objectives in the system-level plans. Some examples of verification objectives the RSC developer cannot typically satisfy and the integrator or applicant must address are:

- Integration;
- Software integration testing;
- Hardware-software integration testing;
- Requirements-based test coverage;
- Timing analysis;
- Memory analysis;
- Stack analysis;
- Data coupling analysis;
- Control coupling analysis;
- Robustness testing;
- Partitioning and other protection mechanisms for integrity validation; and
- Any installation-specific testing, such as system bench testing, aircraft ground and flight testing, and flight test pilot's and human factors specialist's evaluations of flight deck effects.

(2) Some common reverification questions to consider are (not an exhaustive list):

- How much reverification is required if a different compiler type or optimization is used?
- How much reverification is required if a different target environment (microprocessor, memory management unit, timers, memory, input/output devices, databuses, and so on) is used?
- How are data coupling and control coupling analyses performed in the new system for the entire application?
- What reverification or other verification is required to integrate the RSC with other software components into the overall system?
- How much structural coverage analysis should be repeated if the target system changes?
- How much reverification is needed for run-time and compiler libraries?
- If a new target environment is used, what kinds of resource issues exist? For example, are there non-deterministic, dynamic memory allocation algorithms with the RSC that could create resource issues (such as

memory and execution time) in the new target environment? If resource issues do exist, how will reverification be carried out?

- If a new target environment is used and structural coverage changes (that is, there are different unreached code sections), how will the unreached code be addressed? How will it be assured deactivated code cannot be inadvertently activated in the new system?

c. Interface. The RSC developer must provide interface data. These data must clearly define what the integrator or applicant must do to ensure the RSC will perform according to its requirements. Typical items included in interface data are (not an exhaustive list):

- Configuration parameters;
- Restrictions on tools;
- Additional verification activities;
- Memory and timing requirements;
- External resources required by the RSC for proper functioning and performance;
- Definition of the communication mechanisms between the RSC (and other software programs) and the communication protocols with hardware components;
- Accessible variables and their characteristics;
- Variables and data required from the system and their characteristics (for example, inputs to RSC);
- Bus and input/output ports and devices; and
- Access mechanisms.

d. Partitioning and Protection. Although partitioning and protection will most likely be a function at the system level, the RSC itself may require some partitioning and protection. For example, some maintenance code may be at a different software level than the operational flight program for the RSC. Sometimes the RSC might have specific protocols that help protect and partition it. The integrator, applicant, and certification authorities should document and evaluate these protocols.

e. Data Coupling and Control Coupling Analyses. The integrator or applicant must:

(1) Address data coupling and control coupling to ensure they have fully identified and verified all potential side effects of data modifications.

(2) Analyze each side effect to ensure it does not adversely affect functionality or performance.

(3) Evaluate how much the RSC and other integrated software and hardware components depend on data and control interchanges. For example, all modifications of RSC data should only occur at defined interfaces where the RSC can fully control the data behavior.

f. Using Qualified Tools. If qualified tools are used to develop or verify the RSC, the RSC developer must consider reuse of those tools during RSC development and acceptance.

Section 12.2 of RTCA/DO-178B and chapter 9 of Order 8110.49 provide more information on the tool qualification process and the supporting documentation.

(1) When qualified tools are used to develop or verify an RSC, the Tool Qualification Plan and the Tool Accomplishment Summary (or PSAC and SAS for verification tools) must document portions of the tool qualification the applicant must complete. For example, test procedures and cases might have some target dependencies, and the integrator or applicant must perform more verification activities.

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

(2) The RSC developer must provide the following tool qualification data to the applicant for all tools used in getting acceptance of the RSC:

(a) Tool plans;

(b) Tool Operational Requirements; and

(c) The Tool Accomplishment Summary. For some verification tools, the RSC SAS may include the Tool Accomplishment Summary.

(3) All tool data not listed in paragraph 11f(2) of this AC must be available for review by the applicant and certification authority (and authorized designees), as needed, to support continued airworthiness.

g. Deactivated Code. Because RSCs often have functions in the initial application that they will not use in the subsequent applications, deactivated code is a common reuse issue. If implemented, the RSC developer must consider deactivated code during the development of the RSC (for example, the RSC user's guide may define how the user deactivates functions). The applicant or integrator must also consider the approach for implementing deactivated code in their overall system to ensure they comply with RTCA/DO-178B's guidance on deactivated code. In the life cycle data, the RSC developer – and the applicant or integrator – must clearly identify any information about deactivated code and the associated deactivation mechanisms.

h. Traceability. Several RTCA/DO-178B objectives address the traceability of system requirements, high-level requirements, low-level requirements, derived requirements, code, and test cases and procedures. When multiple stakeholders and multiple components are involved, this traceability becomes more difficult. The integrator or applicant must address and maintain traceability between the RSC, the system software, and the system.

i. Robustness. Since the RSC is developed for various applications, it must be able to anticipate out-of-range data or unexpected input (that is, it must be robust). Stakeholders must document how they plan to address the robustness of the RSC. Developers, integrators, and applicants must verify the robustness through robustness tests, while developing and integrating the RSC.

12. CHANGES TO REUSABLE SOFTWARE COMPONENTS.

a. RSCs will likely change. When an RSC is changed, the original reuse status will no longer apply to the changed component (that is, the acceptance letter cannot be used for the modified RSC). If the stakeholders want to change a previously accepted RSC, they must modify the software component using the guidelines of this AC (see paragraphs 12b through 12f) and RTCA/DO-178B, Section 12.1. They also must reaccept the software component as part of a project.

b. When an RSC is changed, the appropriate stakeholders must perform a change impact analysis to identify the changed and affected aspects of the software. The change impact analysis should follow a defined process. This process should determine the potential impact of the change on continued operational safety of the aircraft and the impact of the change to the previously accepted RSC. The change impact analysis should address the following items, as applicable; this list is not all-inclusive and depends on the product being modified:

(1) Traceability analysis identifies areas the software change could affect. This includes the analysis of affected requirements, design, architecture, code, testing, and analyses, as described below:

(a) Requirements and design analysis identifies the software requirements, software architecture, and safety-related software requirements the change affects. Also, the analysis identifies any other features or functions being implemented in the system, ensures added functions are appropriately verified, and ensures the added functions do not adversely affect existing functions.

(b) Code analysis identifies the software components and interfaces the change affects.

(c) Test procedures and cases analysis identifies specific test procedures and cases that will need to be reexecuted to verify the changes and any potential impacts of the changes. This analysis also identifies and develops new or modified test procedures and cases (for added functionality or previously deficient testing), and ensures the changes do not cause any adverse effects. Regression tests may verify the absence of adverse effects. Perform these tests at a hierarchical level appropriate for the software levels of the changed software. Examples of regression tests are aircraft flight tests, aircraft ground tests, laboratory system integration tests, simulator tests, bench tests, hardware and software integration tests, and module tests.

(2) Memory margin analysis ensures memory allocation requirements and acceptable margins are maintained.

(3) Timing margin analysis ensures the timing requirements, central processing unit task scheduling requirements, system resource contention characteristics, interface timing requirements, and acceptable timing margins are maintained.

(4) Data flow analysis identifies changes to data flow and coupling between components and ensures there are no adverse impacts.

(5) Control flow analysis identifies changes to the control flow and coupling of components and ensures there are no adverse impacts.

(6) Input/output analysis ensures the changes have not adversely affected the input and output (including bus loading, memory access, and hardware input/output device interfaces) requirements of the product.

(7) Development environment and process analyses identify any changes, which may adversely affect the software application or product (for example, compiler options or versions and optimization change; linker, assembler, and loader instructions or options change; or software tool change).

(8) Operational characteristics analysis ensures changes (such as changes to gains, filters, limits, data validation, performance, interrupt and exception handling, and fault mitigation) do not result in adverse effects.

(9) Certification maintenance requirements (CMR) analysis determines if new or changed CMRs are necessitated by the software change.

(10) Partitioning or protection analysis ensures the changes do not affect any protective mechanisms incorporated in the design.

c. The change impact analysis should determine if the change to the RSC could adversely affect safe operation of the system or product. The following are examples of areas that could have an adverse impact on installation, safety, operations, functionality, or performance:

(1) Safety-related information is changed. For example:

(a) Previous hazards, identified by the system safety assessment, are changed.

(b) Failure condition categories, identified by the system safety assessment, are changed.

(c) Software levels are changed, particularly if the new software level is higher than the previous level.

(d) Safety-related requirements, identified by the system safety assessment, are changed.

(e) Safety margins are reduced.

(2) Changes to operational or procedural characteristics of the aircraft that could adversely affect flight safety. For example:

(a) Aircraft operational or airworthiness characteristics are changed.

(b) Flight crew procedures are changed.

(c) Pilot workload is increased.

(d) Situational awareness, cautions, warnings, and alerts are changed.

(e) Displayed information for making flight decisions is changed.

(f) Assembly and installation requirements are changed.

(g) Equipment interchangeability or interoperability with other equipment is changed.

(h) CMRs are changed or added.

(3) New functions or features are added to the existing system functions that could adversely affect flight safety or operations.

(4) Processors, interfaces, and other hardware components or the environment are changed in a way they could adversely affect safety, operations, functionality, or performance (see RTCA/DO-178B, Section 12.1.3).

(5) Software life cycle data (requirements, code, and architecture) are significantly changed in a way that could adversely affect safety, operations, functionality, or performance. For example:

(a) Changes to software requirements, design, architecture, and code components (especially those affecting safety-related functions, partitioning, redundancy, or safety monitors).

(b) Changes to code (source, object, and executable object) components that perform a safety-related function or changes to a component providing input to another component that performs a safety-related function. For this AC, a safety-related function is one that could potentially cause or allow a major, hazardous, or catastrophic failure condition to go undetected.

(c) Changes to characteristics of the development environment affecting the executable object code.

(d) Changes to memory allocation requirements so memory margins are adversely affected (for example, less than 5 percent margin remaining).

(e) Changes to timing requirements so timing margins are adversely affected (for example, margins are unpredictable or less than 10 percent margin remains).

(f) Changes to input/output requirements (such as bus loading) so input or output performance is adversely affected (for example, less than 5 percent margin remains).

(g) Data and control coupling characteristics are adversely affected (for example, to the extent more than 50 percent of the coverage analysis must be redone).

(h) Changes to interface characteristics.

d. Also, the following items should be identified in the change impact analysis:

(1) Updates ensure the software changes are incorporated in the appropriate software life cycle data, including requirements, design, architecture, source and object code, and traceability.

(2) Verification activities verify changes and ensure changes do not adversely affect the system. The change impact analysis should cover how to verify changes that could adversely affect safe operation of the system or aircraft, so the changed and unchanged software will continue to satisfy their requirements for safe operation. These verification activities may include reviews, analyses, regression testing, requirements-based testing, robustness testing, flight testing, reevaluation of existing analyses, reexecution of existing tests, and new test procedures and cases (for added functionality or previously deficient testing).

e. When the applicant or integrator changes the RSC without the RSC developer's assistance, that integrator or applicant becomes responsible for satisfying the applicable RTCA/DO-178B objectives for the RSC and all other software components of the system.

f. The RSC developer, users of the RSC, and the appropriate certification authority must coordinate changes to an RSC that result from an airworthiness directive (AD). Coordination will help determine how the AD applies to other projects. An AD issued on any system containing an RSC may invalidate that RSC as reusable.

13. CONCURRENT USE OF AN RSC.

a. Sometimes an RSC may be developed for use by concurrent projects. The development of the RSC and the multiple applications using the RSC may progress at the same time. In this situation, the RSC developer must create a "Reuse Plan" (or equivalent document) which typically includes:

(1) Known applications and projects that will use the RSC (including the first applicant). Because developers may not know all future users when they write the Reuse Plan, the plan should document plans, procedures, and policies for working with future users and certification authorities.

(2) The schedule for the multiple applications and projects.

(3) Statement of the intent to follow this AC.

(4) A proposed reuse approach, based on this AC's guidance and the specific project needs. The Reuse Plan should thoroughly address this AC. The reuse approach should also propose a way to use FAA and designee resources efficiently. For example, shared reviews and review reports may help to optimize resources of FAA and applicants.

(5) A list of all data items (with specific configuration identification) being developed for each user.

(6) A summary of which data items will be the same for all integrators or applicants and which data items are user-specific.

(7) An explanation of data items that differ among users. These may not be suitable for reuse.

(8) An approach for managing configuration of the RSC with multiple users. To gain reuse credit, all users must be using the same RSC configuration.

(9) A list of affected applicants and certification offices. (**NOTE:** Sometimes the list of applicants may be proprietary data that can only be shared with the certification authority. Therefore, the list of affected applicants may be in a separate document to share with certification authorities only.)

(10) A description of how to enable users to use the product correctly (for example, a user's guide or interface document).

(11) A description of how to keep the users up-to-date during the development and deployment of the product. For example, describe how to inform the integrators or applicants of problems with the RSC, potential safety issues, changes to RSC requirements and the user's guide, and other relevant reporting processes.

(12) A description of how to address potential changes to the RSC.

b. The RSC developer must coordinate the Reuse Plan with all appropriate certification authorities, applicants, and integrators. All stakeholders must agree on the approach for concurrently using the RSC. Typically, the focal point for the Reuse Plan is the FAA office that will likely have first approval of a project using the RSC.

14. RELATED DOCUMENTS.

a. Code of Federal Regulations (CFR). Title 14 CFR parts 21, 23, 25, 27, 29, 33, and 35 are relevant to this AC. Copies of the CFR parts are available from the FAA website at http://www.airweb.faa.gov/rgl.

b. FAA Advisory Circulars (AC). AC 20-115B, *RTCA, Inc., Document RTCA/DO-178B*, dated January 11, 1993, offers a way to get FAA approval of software. This RSC AC guides applicants on using RTCA/DO-178B as their means of showing compliance to the regulations for software components. This AC supplements RTCA/DO-178B and AC 20-115B, for accepting compliance based on some RTCA/DO-178B objectives for individual components of a system's software application and functions. If an applicant proposes a means of compliance other than RTCA/DO-178B, the FAA will decide if this AC applies and if other policy or guidance is warranted. You can get copies of this AC, AC 20-115B, and other ACs from the FAA website at http://www.airweb.faa.gov/rgl.

c. FAA Policy Documents. FAA Order 8110.4, *Type Certification* (as amended), and Order 8110.49, *Software Approval Guidelines*, are relevant to this AC. You can get copies of orders from the FAA website at http://www.airweb.faa.gov/rgl.

d. RTCA, Inc. Documents. You may buy copies of RTCA documents from RTCA, Inc., 1828 L Street, NW, Suite 805, Washington, D.C. 20036. Alternatively, you may buy copies on-line at http://www.rtca.org/. RTCA documents relevant to this AC are:

(1) RTCA, Inc., Document RTCA/DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*, dated December 1, 1992.

(2) RTCA, Inc., Document RTCA/DO-248B, *Final Report for Clarification of DO-178B Software Considerations in Airborne Systems and Equipment Certification*, dated October 12, 2001.

e. Society of Automotive Engineers (SAE) Documents. You may buy copies of SAE Aerospace Recommended Practices (ARP) from SAE, 400 Commonwealth Drive, Warrendale, PA 15096-0001. Or, you may buy copies on-line at http://www.sae.org/. The following SAE ARPs are relevant to this AC:

(1) ARP4754, *Certification Considerations for Highly-Integrated or Complex Aircraft Systems*.

(2) ARP4761, *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.*

Susan J.M. Cabler
Acting Manager, Aircraft Engineering Division
Aircraft Certification Service

APPENDIX 1. DEFINITIONS OF TERMS

The RTCA/DO-178B Annex B definitions and the following definitions apply to this AC:

a. Acceptance is credit the FAA grants for fully or partially meeting RTCA/DO-178B objectives for an RSC. The FAA shows acceptance by issuing an RSC acceptance letter.

b. Access mechanism is the manner in which a software component performs its intended function. This includes invocation mechanisms and data flow to and from the component. This is typically part of the interface description data.

c. Applicant is the manufacturer seeking certification of the product or authorization of the equipment. The applicant may be applying for a TC, STC, ATC, ASTC, or TSO authorization.

d. Certification authority is the organization or person in the state or country, who certifies compliance with the requirements. The certification authority is typically the FAA or foreign certification authority engineer.

e. Credit is compliance to one or more RTCA/DO-178B objectives supported by RTCA/DO-178B software life cycle data. This compliance shows the equipment meets the certification basis and may receive a certificate. This AC refers to three types of credit:

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

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

(3) No credit – does not meet the RTCA/DO-178B objective, and the applicant or integrator must complete the objectives for compliance.

f. Designee is a person authorized to make compliance findings on the FAA's behalf for the specific project.

g. First use of RSC is the first acceptance of an RSC in a certification project.

h. In-service experience is experience gained while using the RSC in a certificated aircraft or engine.

i. In-service problem report is the documentation of a problem or difficulty discovered during an in-service experience.

j. Integrator is the manufacturer, who integrates the RSC into the target computer and system with other software components.

k. Interface description data identify the interface details of the RSC. The RSC developer gives the data to the integrator or applicant. The interface description data should

clearly define what the integrator or applicant must do to ensure the RSC will perform per its approval basis.

l. Maintenance code is code in an airborne computer-based system that interfaces with an onboard maintenance computer or computer used by maintenance personnel. This code usually reports to the maintenance computer any problems detected during normal operations.

m. Reusable software component (RSC) is the software, its supporting RTCA/DO-178B software life cycle data, and other 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.

n. RSC developer is the manufacturer of the RSC.

o. RSC user is an integrator or applicant who uses the RSC.

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

q. Software characteristics include the Executable Object Code size, timing and memory margins, resource limitations, and the means of measuring each characteristic (see Section 11.20(d) of RTCA/DO-178B).

r. Software component is some part of the airborne system's software. It usually performs a specific function in the system.

s. Software life cycle data are data produced during the software life cycle to plan, direct, explain, define, record, or provide evidence of successful completion 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.

t. Stakeholders are all the persons and groups involved in the development, integration, and acceptance of the RSC. Stakeholders in this AC are the RSC developer, integrator, applicant, and certification authority. One or more manufacturers may assume the roles of the RSC developer, integrator, and applicant.

u. Subsequent use of RSC is the follow-on use of an accepted RSC. That is, it is not the first use of the RSC.

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

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

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

y. User – see "RSC user" above.

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

APPENDIX 2. ACRONYMS

The following acronyms are used in this AC:

AC	Advisory Circular
AD	Airworthiness Directive
ARP	Aerospace Recommended Practice
ASTC	Amended Supplemental Type Certificate
ATC	Amended Type Certificate
CFR	Code of Federal Regulations
CMR	Certification Maintenance Requirement
CSTA	Chief Scientific and Technical Advisor
FAA	Federal Aviation Administration
PSAC	Plan For Software Aspects Of Certification
RSC	Reusable Software Component
SAE	Society of Automotive Engineers
SAS	Software Accomplishment Summary
SCI	Software Configuration Index
SCMP	Software Configuration Management Plan
SDP	Software Development Plan
SQAP	Software Quality Assurance Plan
STC	Supplemental Type Certificate
SVP	Software Verification Plan
TC	Type Certificate
TSO	Technical Standard Order

APPENDIX 3. SAMPLE FORMAT FOR RSC DEVELOPER'S TABLE

DO-178B Obj #	Objective Description	Credit Sought	Assumption	Means of Compliance for the Objective	Activities Remaining For Integrator or Applicant
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.	Note 1	Note 2	Note 3	Note 4
1-3	Software life cycle environment is defined.	Note 1	Note 2	Note 3	Note 4
1-4	Additional considerations are addressed.	Note 1	Note 2	Note 3	Note 4
1-5	Software development standards are defined.	Note 1	Note 2	Note 3	Note 4
and so on					

NOTE 1: Include if seeking FULL, PARTIAL, or NO credit for the RSC. See Section 5d(1) of this AC.

NOTE 2: List all assumptions made for the credit claim. See Section 5d(2) of this AC.

NOTE 3: List data that documents the compliance to this objective. See Section 5d(3) of this AC.

NOTE 4: List the activities remaining for the integrator or applicant to complete the objective. This should be in enough detail that the integrator or applicant and the certification authority can clearly understand what remains for the overall acceptance of the system using the RSC. See Section 5d(4) of this AC.