Certification Authorities Software Team (CAST)

Position Paper CAST-13

Automatic Code Generation Tools Development Assurance

Completed June 2002

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

Automatic Code Generation Tools Development Assurance

Abstract:

This paper clarifies DO-178B/ED-12B section 12.2.1.b, as it applies to automatic code generation (ACG) tools. DO-178B/ED-12B proposes that the software level of a development tool should be considered at least the same as the software level of the airborne system's software application the tool is being used to develop, **unless** the applicant can justify a reduction in software level of the tool to the certification authority. According to DO-178B/ED-12B:
"*A reduction in a tool's software level can be based upon the significance of the software verification process activity to be eliminated, reduced or automated, with respect to the entire suite of verification activities. This significance is a function of:*
-the type of software verification process activity to be eliminated, reduced or automated …
-the likelihood that other verification activities would have detected the same errors."

This paper proposes a list of candidate objectives in DO-178B/ED-12B that could potentially be alleviated, when applicants are qualifying Automatic Code Generation Tools, provided that the applicants supply relevant rationale and justification for each objective's alleviation. The paper also provides a road map to potentially reduce the ACG tool's software level relative to the level of the airborne software.

The approach proposed in this paper is only one of many potential approaches. Other approaches would need to be coordinated with the appropriate certification authorities.

Key words:

Tool qualification, autocode generators, development tools, code, verification, software level, objectives, tool software level

1 INTRODUCTION

This paper clarifies DO-178B/ED-12B section 12.2.1.b, as it applies to automatic code generation (ACG) tools. DO-178B/ED-12B proposes that the software level of a development tool should be considered at least the same as the software level of the airborne system's software application the tool is being used to develop, **unless** the applicant can justify a reduction in software level of the tool to the certification authority. According to DO-178B/ED-12B:

"*A reduction in a tool's software level can be based upon the significance of the software verification process activity to be eliminated, reduced or automated, with respect to the entire suite of verification activities. This significance is a function of:*

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

-the type of software verification process activity to be eliminated, reduced or automated …
-the likelihood that other verification activities would have detected the same errors."

This paper proposes a list of candidate objectives in DO-178B/ED-12B that could potentially be alleviated, when applicants are qualifying Automatic Code Generation Tools, provided that the applicants supply relevant rationale and justification for each objective's alleviation. The paper also provides a road map to potentially reduce the ACG tool's software level relative to the level of the airborne software.

The approach proposed in this paper is only one of many potential approaches. Other approaches would need to be coordinated with the appropriate certification authorities.

2 BACKGROUND

According to DO-178B/ED-12B chapter 12.2.1.b, the software level of a development tool should be at least the same as that of the airborne system's software being generated by the tool, if the code generated by that tool is not verified as defined in DO-178B/ED-12B Section 6. Since an ACG is a development tool, this implies that ACG tools need at least the same level of assurance as the airborne software they produce.

DO-178B/ED-12B chapter 12.2.1.d recognizes that the objectives of the tool's software verification are different from those of the airborne software. It recognizes implicitly, therefore, that not all verification objectives of DO-178B/ED-12B may be fully applicable to an ACG tool. It provides a list of verification activities for software development tools that is a subset of those suggested for airborne software.

Additionally, according to DO-178B/ED-12B chapter 12.2.1.b, the applicant can justify a reduction in software level of the tool to the certification authority. A reduction in a tool's software level can be based upon the significance of the software verification process activity to be eliminated, reduced or automated, with respect to the entire suite of verification activities. This significance is a function of:

- *The type of software verification process activity to be eliminated, reduced or automated*
- *The likelihood that other verification activities would have detected the same errors*."

Therefore, the accurate way to determine the necessary qualification process of the ACG tools (the applicable verification objectives) is somewhat debatable and requires

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

clarification. There is a need to identify the verification objectives that may not be applicable to the ACG tools and rationale for why they may not be applicable, and to justify why other applicable objectives can be satisfied without reducing the level of assurance for the airborne software application.

3 DISCUSSION

3.1 ACG context presentation

ACG tools are applications that execute on ground computers using commercial off-the-shelf (COTS) operating systems.

ACG tools may directly translate the software specification (low-level and/or high-level requirements) into source code, thereby potentially eliminating or automating several software development and verification activities. ACG tools are generally composed of two entities:

- Entity 1: The library of elementary symbols (e.g., code primitives or basic functions) that contain basic symbols that contain the source code associated with implementing the function of each elementary symbol.

- Entity 2: The architect (e.g., tool logic and program constructor) that reads the software specification and selects the association to the elementary symbol, and may insert the corresponding source code for each selected symbol.

The qualification approach for each of these entities may vary.

When tools do not eliminate, reduce, or automate software processes necessary to be performed on the airborne software, those tools do not need to be qualified.

When tools eliminate, reduce, replace, or automate life cycle process activities on the airborne software, without their output being verified as specified in Section 6 of DO-178B/ED-12B, the tools should be qualified.

Once it is determined that a development tool needs to be qualified, the software level of the tool to be qualified needs to be determined.

In the case where there is no justification to reduce the software level of the tool, the tool should be qualified to the same software level and objectives as that of the airborne software.

In the case where there is a justification to reduce the tool's software level, the verification objectives and activities for the reduced software level should be satisfied.

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

Justification for reducing the software tool's level should be evaluated and analyzed addressing the following issues:

- What are the possible consequences of a tool's software level reduction on the airborne software product itself (for example, possible undetected errors in the airborne software and implications on the system)?
- What are additional, complementary verification activities proposed for the tool's output or on the airborne software to detect these errors, or mitigate software effects of these errors on the system?
- What analysis supports the proposed reduction of the tool qualification objectives?

3.2 ACG tool qualification level and objectives clarification for potential alleviation

According to Section 12.2.1.b of DO-178B/ED-12B, a development tool should be qualified to the level of the embedded software. The intention of this statement is explained in section 12.2.1.a: the software development processes for the tool should meet the same objectives as the software development processes of the airborne software.

The technical objective is that ACG tools should not introduce unintended or erroneous code in the embedded software. The ACG tool behavior should be deterministic; i.e., the produced code should not differ when the input data does not.

More precisely, the primary issue for an ACG tool is the production of source code that does not comply with its requirements, but can still be compiled without any error detected and is executable. For airborne software, the same event can occur, but it's not the only one. Halts during execution, overflows, variations in time response, hardware and software incompatibilities, hardware failures, unbounded recursive algorithms, bad stack usage, resource contention, tasks conflicts, bad interaction with others systems, etc. are examples of issues which may jeopardize flight safety, if they appear in aviation software. However, these types of errors may not have any influence on the flight safety, if they occurred in the ACG tool that generated the aviation software.

Therefore, the following list better describes how to demonstrate compliance to DO-178B/ED-12B for a typical ACG tool (i.e., one that uses the two entities described above):

- The operating system on which the ACG tool executes should be considered. The applicant should demonstrate that the tool is designed and developed in such a way that erroneous functioning of the operating system cannot produce unintended or erroneous code (e.g., showing tool operational requirements in abnormal conditions). Neither can it jeopardize determinism properties of the tool (i.e., the produced code should not differ when the input data does not, as previously mentioned).

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

- The ACG tool entity 1 (symbol libraries) should be shown to comply with all the objectives of DO-178B/ED-12B.

- The ACG tool entity 2 (program architect), should satisfy all of the objectives of DO-178B/ED-12B (for the applicable level), except for a list of candidate objectives, proposed according to DO178B/ED-12B chapter 12.2.1.d, and any additional field experience. These candidate objectives are objectives that applicants could consider for alleviation or removal. The candidate objectives for alleviation or removal are: A-3 (3), A-4 (3, 10, 11), A-5 (6), A-6 (partial 2, partial 4, 5), and A-7 (5). All other objectives should be applied, unless further justification is provided. The table below summarizes the DO-178B/ED-12B reference, alleviation rationale, and suggested demonstration approach for each of the candidate objectives. Each applicant should provide rationale and justification for alleviation or removal of each objective, but this provides a starting point:

Obj	Ref.	Rationale	Demonstration requirements
A-3 (3)	6.3.1.c	This objective could be alleviated. ACG tool incompatibility with the target computer will likely be obvious as the ACG tool will likely not execute nor would the ACG tool likely satisfy its Tool Operational Requirements. The hardware and software incompatibilities are managed by the operating system using hardware resources (e.g., CPU, memory, I/O devices, etc. installed in a PC or a Unix Workstation). Therefore, incompatibility problems may be obvious and readily detected.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e., production of unintended or erroneous code).
A-4 (3)	6.3.2.c	This objective could be alleviated. ACG tool incompatibility with the target computer will likely be obvious as the ACG tool will likely not execute nor would the ACG tool likely satisfy its Tool Operational Requirements. The hardware and software incompatibilities are managed by the operating system using hardware resources (e.g., CPU, memory, I/O devices, etc. installed in a PC or a Unix Workstation). Therefore, incompatibility problems may be obvious and readily detected.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e., production of unintended or erroneous code).
A-4 (10)	6.3.3.c	This objective could be alleviated. ACG tool software architecture may not be defined at the ACG tool level. An ACG tool is application software running on a workstation operating system with its own task management, memory management and scheduling. Incompatibility problems between the ACG tool's architecture	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e. production of unintended or erroneous code).

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

		and the computer may be obvious and readily detected, and the ACG tool would likely not satisfy its Tool Operational Requirements if incompatibilities existed.	
A-4 (11)	6.3.3.d	This objective could be alleviated. The recursive algorithm is allowed if the unbounded recursive algorithm is detected and cannot lead to production of unintended or erroneous code. An infinite loop has different implications for a tool's execution than it does for airborne software, and would likely be detected during tool execution; also, the operator could stop it. Nevertheless, the software architecture is partially verified during Tool Operational Requirements testing.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e. production of unintended or erroneous code).
A-5 (6)	6.3.4.f	This objective remains applicable except for the worst-case execution timing, stack usage, resource contention, task or interrupt conflict, which could all be possibly alleviated. Worst-case execution is not an issue for the ACG tool execution as it only impacts tool availability. Stack usage is not an issue, if the data used to produce source code are not corrupted. Resource contention is not an issue if data used to produce source code are not corrupted. Task or interrupt conflict may not be an issue, if it only impacts ACG tool real-time availability and cannot corrupt data used to produce source code.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e. production of unintended or erroneous code).
A-6 (partial 2 and partial 4)	6.4.2.2 b	This objective could be potentially alleviated. Any system initialization problems will likely be obvious and result in temporary or permanent ACG tool unavailability or the need to restart the tool or operating system. Also, the abnormal conditions will likely be obvious.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e. production of unintended or erroneous code).
A-6 (partial 2 and partial 4)	6.4.2.2 c	This objective could be alleviated. There is no data coming from external systems. Input data may be formalized requirements specifications and the output data is source code. An ACG tool is not a system, but an application running on an operating system. If the formalized specification contains errors or defects, the tool should detect them.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e., production of unintended or erroneous code). The applicant should also demonstrate that any error or defect in formalized specification would be detected by the tool.
A-6 (partial 2	6.4.2.2 e	This objective could be alleviated. The operating system is managing real-time	Applicant justification is needed. It should be demonstrated that the

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

and partial 4)		resources, and time frame exceedances should only lead to temporary or permanent unavailability of the ACG tool.	objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e., production of unintended or erroneous code).
A-6 (partial 2 and partial 4)	6.4.2.2 f	This objective could be alleviated. An ACG tool generally does not have real-time constraints, as do some airborne software applications. It is a ground-based application running on an operating system, which has its own time and task management schemes. Problems in this area should only lead to temporary or permanent unavailability of the ACG tool.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e., production of unintended or erroneous code).
A-6 (5)	6.4.3a	This objective is applicable. Nevertheless, activities that check for real-time properties (such as, memory overflow, failure to detect execution time requirement's anomalies, inability of built-in test to detect failures, and stack overflows) may not be applicable for ground-based software applications.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e., production of unintended or erroneous code).
A-7 (5)	6.4.4.2.b	This objective remains applicable except for additional verification on the object code of the tool (for Level A tools). Problems in object code not directly traceable (due to usage of an OS for example) to the ACG tool's source code are not an issue as they occur on the ground, may affect only real-time availability of the ACG tool, cannot corrupt data used to produce source code, and cannot lead to producing unintended or erroneous code.	Applicant justification is needed. It should be demonstrated that the objective alleviation can only cause ACG tool availability problems and cannot lead to ACG tool integrity problems (i.e., production of unintended or erroneous code).

Potential errors caused by the objectives alleviated should be demonstrated as not being a potential source of errors in the airborne software. The potential errors should be shown as detectable by other means and should not lead to production of erroneous code. Remaining objectives should be applied at the same level as the airborne software level, unless the applicant can justify that an objective or activity alleviation cannot be the cause of errors in the airborne application.

Each time an objective is alleviated, the alleviation of the objective should be described and justified.

3.3 ACG software level reduction and road map proposal

According to the previous section, DO-178B/ED-12B objectives that remain applicable for an ACG tool should be applied at the same software level as the embedded

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

airborne software's level, unless the applicant can justify reduction of the tool's software level or proposes other means, such as additional verification activities to satisfy an objective. Therefore, applicants could propose their reduction, justification and additional means for alleviating further objectives than the ones proposed in Section 3.2 above for ACG tool entity 2.

If it can be shown that software functions included in ACG tool entity 2 cannot contribute to the production of non-deterministic, unintended, and erroneous code, the tool's software level for these items could potentially be reduced to the associated failure condition classification to which the tool's functions could contribute, if the tool's functions can be partitioned, and independently verified.

If it cannot be shown that a software function included in ACG tool entity 2 cannot contribute to the production of non-deterministic, unintended and erroneous code, all objectives applicants propose to satisfy for the tool should be satisfied to the same level as the airborne software application which the ACG tool generates. Any possible consequences of further reduction of the ACG tool software level or objectives' alleviation (further to what is proposed in section 3.2) should be evaluated for their potential impact on the airborne software product, to justify the final ACG tool's software level and qualification claimed.

- A: Potential errors resulting from the ACG tool software level reduction can be detected by other means (including additional verification performed on the code produced). The equivalency of other means should be assessed on the ability of the method to detect exactly the same kind of errors as the verification objective that is being alleviated would have detected.

- B: Potential errors and failure conditions in the airborne software as a result of the ACG tool's software level reduction or objectives' alleviation cannot contribute to the failures of the airborne software. In this case, the ACG tool software level could perhaps be lower than the software level of the airborne software. The applicant can propose a reduction based on a safety assessment, software architecture, tool partitioning, additional verification, and/or by other means, for consideration by the certification authority on a case-by-case basis.

The flowchart in Appendix A illustrates the roadmap described in this section.

4 CAST POSITION

This paper proposes a list of candidate objectives in DO-178B/ED-12B that could potentially be alleviated, when applicants are qualifying ACG tools, provided that the applicants supply relevant rationale and justification for each objective's alleviation.

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

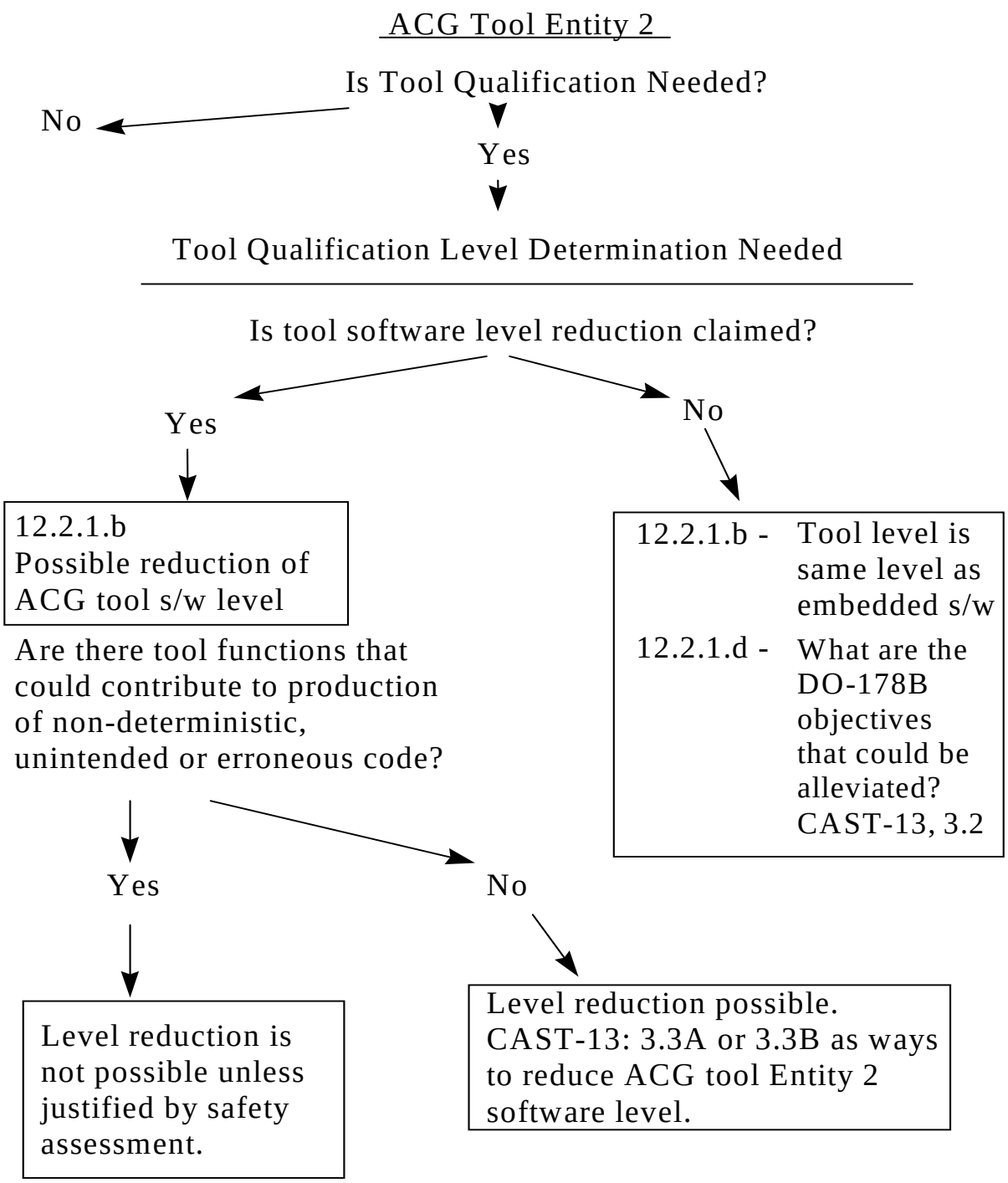
The paper also provides a roadmap to potentially reduce the ACG tool's software level relative to the software level of the airborne software.

CAST considers the proposed approach, if properly applied by applicants, to be an acceptable way to alleviate DO-178B/ED-12B objectives for ACG tool qualification, as well as to potentially reduce the ACG tool's software level relative to the software level of the airborne software.

The approach proposed in this paper is only one of many potential approaches. Other approaches would need to be coordinated with the appropriate certification authorities.

NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

APPENDIX A



NOTE: **This position paper has been coordinated among the software specialists of certification authorities from the United States, Europe, and Canada. However, it does not constitute official policy or guidance from any of the authorities. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**