

Complex Electronic Systems Developer Tool "dBricks"
Overview

dBr.0003.05.en.01

Table of contents

1	List of Abbreviations, Terms and Definitions.....	3
2	Problem Statement	4
3	Purpose.....	5
4	Tool development history	5
5	Tool Description.....	6
5.1	General Description.....	6
5.2	Data Storage Subsystem Architecture.....	6
5.3	Tool Multiuser Features	7
5.4	User Interface Approach	7
5.5	Configuration Control.....	7
5.6	Data Validity Control	8
5.7	Tool Export Features	8
5.7.1	Export of Block Diagrams, Connection Diagrams, Wiring Lists and Schematics, Circuit Diagrams	8
5.7.2	ICD Export	8
5.7.3	Export of I/O Specifications and Software Specifications	8
5.7.4	Export of Data Networks Configuration Files.....	9
5.7.5	Development and Export of Design Documents for Cable Nets of Test Benches and Flight Training Devices	9
5.7.6	Export of Airborne Data Acquisition and Maintenance Systems Configuration Files	9
5.7.7	Export of Configuration Files for IOs of Simulations as Part of Test Bench or Flight Training Devices	9
5.7.8	Export of Analytical Reports	9
6	Tool Deployment Options.....	9
6.1	Cloud-Based Software As Service	10
6.2	Customer-Hosted Server	10
7	Tool Modules.....	10
7.1	Basic Module	10
7.2	Module for Working with Discrete and Analog Buses.....	11
7.3	Module for Working with ARINC 429 Buses	11
7.4	Module for Working with ARINC 825 Buses	11
7.5	Module for Working with ARINC 653 Buses	11
7.6	Module for Working with ARINC 664 (AFDX) Buses	11
7.7	Module for Working with MIL-1553 Buses.....	11
7.8	Configuration Control Module	12
7.9	VHTNG Format Export Module.....	12
7.10	Microsoft Visio Diagrams Export Module.....	12
7.11	API Module	12
7.12	Analytical Reports Module	12
7.13	Toolbox for Development of Test Bench and Flight Simulator Cable Nets	12
7.14	Toolbox for Development of Test Benches and Flight Simulators Configuration Files	12
8	Technical Support and Tool Update Policy.....	12
8.1	Phone Support.....	13
8.2	Email Support	13
8.3	Response to Comments Submitted in Bug & Improvement Tracking System	13
8.4	Upgrade Policy.....	13
9	Conclusion	13

1 List of Abbreviations, Terms and Definitions

Abbreviation / Term	Full Name / Definition
AC	Aircraft
AE	Avionic Equipment / Avionics
AE Component	Part of AE system
API	Application Programming Interface
CAD	Computer Aided Design (system)
CRUD	Create, Read, Update, Delete - four basic functions for working with objects
CN	Cable Net / Cabling
DB	Database
DS	Design Specification (documentation)
EI	Export / Import - one of the main mechanisms for data group add / edit
HIRF	High-Intensity Radiated Fields
ICD	Interface Control Document
Integration	Process of system integration (combining) of subsystems components into one system and ensuring the functioning of subsystems together as a single unified system.
PL	Part List
SW	Software

2 Problem Statement

Last decades innovations in avionics development have provided unprecedented growth of flight safety, convenience of pilots and flights timely departure. But the downside of high-tech intelligent airborne systems is design complexity.

Airborne systems engineers face an extremely difficult task of integrating a huge number of components (which already are complex per se) into a synergy that should work as a single whole. The task is further complicated by the need to provide the highest standards of safety and comply with numerous industry standards.

Development of complex airborne systems cannot do without preparing numerous documents, such as:

- a) Interface control documents;
- b) Block;
- c) Connection diagrams;
- d) Wiring lists and schematics;
- e) Circuit diagrams;
- f) Software input/output design specification;
- g) Network switch configuration tables;
- h) Software design specifications.

Usually the aforementioned documents are big, they contain project implementation details: from wires to digital bus parameters refresh rates. In order for the resulting equipment system to operate fault-free and to be certified by aviation authorities all documents should be consistent and comply with industry standards, regulatory documents, aircraft and avionics requirements.

The challenge of documentation consistency is a consequence of modern avionics complexity in general and the fact that a dozen of departments within the aircraft manufacturer company and dozens of external OEM companies are involved in this documentation development. Organizing effective interaction of all the resources involved in the process is a non-trivial task both from the technical and organizational point of view.

Aircraft (AC) development is governed by AC technical and functional requirements, AC certification requirements (certification basis), applicable state and industry standards, AC manufacturer and OEM's internal standards. All these requirements must be consistent; traceability to them must be controlled (if possible by automatic means).

The initial design data alongside with requirements and regulations includes a large amount of information exchanged between the AC manufacturer and OEMs. These initial data change at different design stages as well as at the stages of testing, trials, certification, and even during the aircraft operation. Controlling the coordinated change of all systems and equipment affected by the initial data changes is one of the aviation equipment development most time-consuming tasks.

Unfortunately, progress in the field of avionics as such has not touched on the avionic equipment (AE) development tools, and engineers are compelled to conduct their projects in an old-fashioned way in spreadsheet and text editors, such as Microsoft Excel and Microsoft Word. These editors, although being convenient and powerful tools for creating documents, are not designed for airborne equipment development, since they do not allow controlling integrity, applying changes throughout the project simultaneously, providing multi-user access to the project being developed, etc.

The result of the absence of specialized tools is the need to perform a huge amount of work on thorough inspection and manual configuration management of documents being developed and modified, which can lead to delays and/or errors.

The existing problem in avionic equipment development processes is recognized by the majority of specialists involved, but the automation tools of the processes described above are not yet available on the market.

3 Purpose

The purpose of the dBricks tool is:

1. Reducing development effort of general complex electronic systems and avionics equipment in particular;
2. Cost reduction for testing and commissioning of systems being developed;
3. Quality improvement for documents being developed.

The tool's purpose is achieved by means of the following processes automation:

1. Avionics equipment configuration management;
2. Ensuring operational coordination between all the stakeholders of development process;
3. Development of AE block diagrams;
4. Development of connection diagrams;
5. Development of wiring lists and schematics;
6. Development of circuit diagrams;
7. Development of interface control documents;
8. Development of flight software (SW) design specifications;
9. Development of data networks configuration;
10. Development of design documentation for cable nets of test benches and flight simulators;
11. Development of configuration files for airborne data acquisition and maintenance systems;
12. Development of configuration files for IOs of simulations as part of test bench or flight simulator;
13. Control of compliance of input data with the requirements of standards and regulatory documents;
14. Analysis of compliance of equipment with the requirements, including but not limited to the following checks:
 - a) Functions source data coverage analysis;
 - b) Failure hazard analysis of equipment and wiring failures;
 - c) Analysis of the bus load and delays in digital data channels;
 - d) Analysis of computational resources use.

4 Tool development history

The dBricks tool is developed on the basis of the vast experience gained during the work on all modern projects of civil airliners in Russian Federation.

The idea of using automation tools for AE development processes emerged during the work on flight instruments of IL-96, TU-204 and IL-114 aircraft. At the same time, the first basic tools for the development of interface control document and wiring lists and schematics were developed.

Later, the tools were further developed and expanded for the automated generation of aircraft circuit diagrams, connection diagrams, interface control documents and other documents used in manufacture, installation and development of the cable net and AE of the SuperJet-100 aircraft (RRJ-95). With the help of these tools, data were also obtained for automated assessment of the stability of the complex against external factors influence including HIRF, and analysis of the aircraft's equipment fail-safety.

Upon completion of the development of the SuperJet-100 a further step was made to connect disparate tools into a single unified system with enhanced functionality. Today the results of these works are widely used in the development and trials of avionic equipment of the MS-21 aircraft.

The dBricks tool is the ideological successor of all the aforementioned stages of automation tools development, which has incorporated all the gained experience and knowledge on the topic.

5 Tool Description

5.1 General Description

The dBricks system is made out of normalized database and means of data input, output and modification.

The tool works on client-server technology. Server side includes network storage and server module. Server module interacts with client module with regard to processing input data and prepares data for visualization by client module.

Client module is a graphical shell intended for displaying data received from server and providing convenient access to the data. dBricks client part of is implemented as a web page accessed through a web browser (e.g. Google Chrome).

5.2 Data Storage Subsystem Architecture

Following the idea of data normalization, the dBricks system considers data describing avionic equipment not as a single entity, but as an object consisting of other objects and connections between them. The objects that make up a project, in turn also consist of smaller objects and connections between them. Using normalized approach completely eliminates the need of repeated entering and storing the same type of data, including when using the same versions of devices in different projects.

Below are three global object families:

1. Common objects: This group contains objects that describe basic concepts, such as bus types, connector types, wire types, data types, parameter units, and so on. Description of these objects does not contain references to other objects, but objects with complex structure can refer to them.
2. Device templates: Device templates are composite objects that describe structure and properties of device types. Usually they describe all devices with the same part number. Device Template can contain the following components in its description:
 - a) Description of the device itself - name, P/N, description, weight, size, manufacturer, etc.;
 - b) List of device connectors - names of connectors, connector block and cable parts types, connector pin layout, materials, manufacturer, etc.;
 - c) List of device ports - device port is a possibility of connecting this device to others, e.g. it can be an output port of the standard ARINC 429. Port properties include name, type, description, etc.;
 - d) List of functions that determine device purpose, indicating the functional parameters necessary for device operation (received and/or transmitted by the device);
 - e) List of individual software elements or partitions (in terms of the ARINC 653 standard) implemented in the device with a list of input/output variables;
 - f) Device port contents - contains a detailed description of the transport layer and configurable port characteristics, e.g. such as ARINC 429 bus transmitted data (word list, refresh rates, parameter mapping to data bits, transmitted data characteristics, etc.). Port contents vary considerably depending on its

type. Port contents can be mapped to functional parameters and/or input/output variables of software elements, allowing linking the data flow requirements at the logical level with the description of the transmission method.

3. Projects - highest level objects describing the design of an avionics complex as a whole. In the first approximation, any project is made up of devices instantiated from device templates and connections between those devices. Device connection description includes:
 - a) Connections between device ports, they form data buses;
 - b) Connections between devices' functional parameters that determine the flow of data at the logical level;
 - c) Description of configuration of complex digital buses, such as ARINC 664, in particular the requirements for the addresses used, the distribution of messages over virtual links (Virtual Link), etc.

5.3 Tool Multiuser Features

The dBricks tool is implemented on client-server architecture, which allows the entire project team to work with the tool simultaneously. This fact is one of the main advantages of the tool. Once the tool is deployed, the most up-to-date information about the status of the project is always available to each project stakeholder. Any changes made by one user automatically become available to everyone. Depending on the Customer's need the tool can be installed in such a way that it will be available not only to the employees of the company, but also to employees of contractor companies. Information security and unauthorized access control is provided by a flexible system of access rights control. The system also provides logging of all user actions, which allows finding the author of each change.

5.4 User Interface Approach

The tool is developed hand in hand with end-users and developers of the AC AE. In the course of development dozens of suggestions from users were accounted in the user interface. The main technical solutions are tested on the real projects data. Tools and speed of the system enable effective work with projects of modern airliners containing hundreds of devices, tens of thousands of wires and hundreds of thousands of parameters.

For the purposes of input and editing of large volumes of data, the tool provides the mechanism of bulk data input/editing, which allows reducing the complexity of the task by orders.

5.5 Configuration Control

During the development of complex AE projects, it becomes necessary to manage the configuration of multiple versions of the project, arising both from the variability of the project itself (for example, the implementation of options), and because of the need to track the generations of the complex developed with different sets of requirements (rig tests version, first flight version, certification version, etc.). dBricks uses configuration management mechanism which allows performing the following functions:

1. Create a baseline version of the device or project configuration template. Once created, the baseline version cannot be changed, which makes it possible to rely on reports generated from that control version for further work.
2. Create new versions of objects based on baseline versions. In the tool terminology this process is called "unfreezing". In the process of unfreezing the tool creates a new exact copy of the baseline version, available for making changes. At the same time several variants of device templates or projects can be created from each baseline version, which allows simultaneous development of several modifications.

3. Compare versions, generate lists of differences.

5.6 Data Validity Control

Using the tool allows to eliminate significant number of errors that are introduced upon data input. The system allows filtering out errors that violate the requirements of regulatory documents, industry standards and customized project limitations. Errors that violate the requirements of regulations and standards include but not limited to attempts to connect different types of buses or attempts to include data in the data exchange description that cannot be physically realized. Errors that violate configurable project limitations may include violation of the naming convention or restrictions on the accidental connection of parameters with different data types.

5.7 Tool Export Features

The tool with a populated database allows you to automate the following processes.

Note: Generally speaking, once the tool is deployed the implementation of these processes is reduced to populating the database. Once the database is populated the resulting documents are created automatically.

5.7.1 Export of Block Diagrams, Connection Diagrams, Wiring Lists and Schematics, Circuit Diagrams

The diagrams listed are the main documents describing the connection of AE devices to each other. The tool allows generating the diagrams in an automated fashion. The result of the export is the completed diagram file in Microsoft Visio format.

5.7.2 ICD Export

ICD development is one of the fundamental task of the entire AE development effort and one of the main purposes of the dBricks tool. The tool allows export of ICD for data exchange implemented through the following types of interfaces:

1. Discrete IOs;
2. Analog IOs incl. proximity sensor signals;
3. Linear Variable Differential Transformer (LVDT) signals;
4. ARINC 429;
5. ARINC 664 (AFDX);
6. ARINC 825;
7. MIL-1553.

In addition to work effort reduction the automated ICD export facilitates elimination of human errors emerging when manually creating the documents.

Note: ICD export format can be customized based on customer requirements. Out of the box dBricks contains export to Microsoft Excel format

5.7.3 Export of I/O Specifications and Software Specifications

In modern projects development of flight software responsible for information input and output is usually produced with specialized equipment manufacturers CAD, based on special format spreadsheets. The presence of complete information on the equipment interaction in the dBricks system makes it possible to prepare the relevant spreadsheets by simply converting the information into the specified format. Through this the possible sources of transfer errors are eliminated and table preparation time is reduced.

Flight software specifications are the documents that fully describe functional requirements of airborne equipment processing modules. The dBricks tool allows generating such documents automatically based on data stored in the database.

5.7.4 Export of Data Networks Configuration Files

In modern data networks such as ARINC 664 the configuration of information flows is described by configuration files that are uploaded to data exchange nodes. Such files can also be automatically generated by the tool based on data put into the database.

Note: Detailed description of the configuration requirements is given in ARINC 664 part 7.

5.7.5 Development and Export of Design Documents for Cable Nets of Test Benches and Flight Training Devices

Traditionally development of cable nets design specifications (CNDS) for test bench and flight training devices (hereinafter referred to as test benches) was a very time-consuming task coming from the large number of errors that appear in the process of converting the aircraft data into documentation adapted to create test benches. Using the dBricks tool allows to:

1. Completely eliminate errors that occur during data conversion;
2. Reduce by 4-5 times the labor cost of CNDS development;
3. Automatically create documentation for test bench cable net update when aircraft design changes are introduced.

5.7.6 Export of Airborne Data Acquisition and Maintenance Systems Configuration Files

The airborne data acquisition systems used in aircraft testing process and the maintenance system are very similar in terms of data exchange. The presence in the dBricks system of complete information about the pick-up points and the formats of the information transmitted by the systems allows to automatically create registration system input devices requirements according to the nomenclature and the number of receivers, connection tables and information formats for its processing

5.7.7 Export of Configuration Files for IOs of Simulations as Part of Test Bench or Flight Training Devices

During test benches and flight training devices (hereinafter referred to as test benches) development there is a need to describe the data exchange of equipment simulators not present on the test benches. Typically test bench simulation systems can be configured through specialized configuration files. Automated creation of such files based on the data entered during the development of ICDs allows to completely eliminate errors in the development of the files, as well as to reduce the labor cost of the process by 95-99%.

5.7.8 Export of Analytical Reports

The presence of integrated information on all aircraft equipment, including support systems (electrical system, hydraulic system, etc.), its location in the aircraft and interconnection allows implementing several important analytical functions:

1. Analysis of power buses load, analysis of equipment failures when one or more power buses are down;
2. Analysis of information buses load, search for common points in case of data transmit/receive failures;
3. Tracing a signal through all the cable net technological elements and data converters, which greatly accelerates the search of problems in aircraft systems development;
4. Analysis of the list of equipment involved in the critical functions of the aircraft;
5. Estimation of the probability of loss of aircraft functions (including critical ones) due to possible failures (Fail-safety analysis of the complex).

6 Tool Deployment Options

Depending on Customers' need the tool can be used in one of the ways listed below.

6.1 Cloud-Based Software As Service

When working with the tool as an Internet service, the Customer gets access to a copy of the software, deployed by the Vendor in the cloud. Access is granted during the period paid by the Customer, and is automatically renewed given that the next period is paid on time. At Customer's request, after switching to the option of installing the tool on the Customer's local server, the work results obtained with the in-cloud tool instance can be transferred to the tool installed on the Customer's local server.

6.2 Customer-Hosted Server

This method involves installing the tool on a server hosted by the Customer. The license term is unlimited subject to the terms of the license agreement. Software technical support and updates are performed during the period of technical support specified in the license agreement. The technical support period can be extended on the terms specified in the license agreement and maintenance contract

7 Tool Modules

Depending on Customer's needs, the tool functionality can be customized by adding/removing the following modules and extension packages:

7.1 Basic Module

The basic module is an integral part of the tool. The module implements the following functionality:

- a) Access rights control;
- b) Creating, reading, updating and deleting (CRUD) of basic elements;
- c) Export/import (EI) of these basic elements;
- d) Manage restrictions on data entered;
- e) Generation of basic reports.

The basic elements for which the CRUD functionality is implemented include:

- a) Bus types;
- b) Cable types;
- c) Connector types;
- d) Data types;
- e) Device templates;
- f) Device templates connectors;
- g) Device templates physical ports;
- h) Device templates functions, including function parameters;
- i) Projects;
- j) Devices as part of projects;
- k) Connections between physical ports of devices in the project;
- l) Connections between function parameters of devices in the project.

EI functionality for basic elements includes EI of:

- a) Parameters of device template functions;
- b) Logic implemented by the device template function;
- c) Device template physical ports;
- d) Project devices;
- e) Connections between function parameters of devices in the project.

The functionality of data input restrictions control allows:

- a) Set restrictions on valid names and indexes of objects;
- b) Specify the auto-generation method for bus names;
- c) Set constraints on connections between variables by data type and units

The basic reports implemented in the module include:

- a) List of project devices;
- b) Project device connections;
- c) Connections of buses of the same type devices in the project;
- d) Parameter path - shows the full path of data from producer to consumer;
- e) Wiring interconnection table.

7.2 Module for Working with Discrete and Analog Buses

The module allows to CRUD information for the following bus types:

- a) Discrete signals of type "Open/Ground";
- b) Discrete signals of type "Open/Voltage";
- c) Analog buses of type "Variable amplitude voltage";
- d) Analog buses of type "Variable resistance";
- e) Analog busses of type "LVDT";
- a) Analog buses of type "Inductive proximity sensor"

The module also implements the export of ICD by buses types listed above.

7.3 Module for Working with ARINC 429 Buses

The module is designed to work with ARINC 429 bus data. The module implements the following functionality:

- b) CRUD of ARINC 429 bus data elements;
- c) EI of the list of ARINC 429 bus data elements;
- d) Report "ARINC 429 Bus Data Exchange".

7.4 Module for Working with ARINC 825 Buses

The module is designed to work with ARINC 825 bus data. The module implements the following functionality:

- a) CRUD of ARINC 825 bus data elements;
- b) EI of the list of ARINC 825 bus data elements;
- c) Report "ARINC 825 Bus Data Exchange".

7.5 Module for Working with ARINC 653 Buses

The module is designed to work with descriptions of flight software data exchange port structures (application/partition ports in terms of the ARINC 653 standard). The module implements the following functionality:

- a) CRUD of flight software application port data elements;
- b) EI of the list of flight software application port data elements;
- c) Reports "Application Port Structure" by the number of types of port structures.

7.6 Module for Working with ARINC 664 (AFDX) Buses

The module is designed to work with descriptions of the structure of networks implemented according to ARINC 664 (AFDX) standard. The module implements the following functionality:

- a) CRUD of ARINC 664 data exchange elements, such as virtual channels, data messages, ES information ports, etc.;
- b) Reports containing information on the structures of ARINC 664 networks.

7.7 Module for Working with MIL-1553 Buses

The module is designed to work with MIL-STD-1553B data. The module implements the following functionality:

- a) CRUD of MIL-STD-1553B bus data elements;
- b) EI of the list of MIL-STD-1553B bus data elements;
- c) Report "MIL-STD-1553B Bus Data Exchange".

7.8 Configuration Control Module

The module implements the functionality described in Section 5.5.

7.9 VHTNG Format Export Module

The module implements the export to the VHTNG standard format.

7.10 Microsoft Visio Diagrams Export Module

The module implements generation of block diagrams and connection diagrams in Microsoft Visio format. For large schematics the module provides the ability to configure the layout of devices on the diagram to simplify reissuing.

7.11 API Module

The module is a set of procedures, structures and constants that allow working directly with the backend of the tool bypassing the user interface. Access through the API allows Customers to use their own automation tools in conjunction with the dBricks tool

7.12 Analytical Reports Module

The module implements export of the reports listed in Section 5.7.8

7.13 Toolbox for Development of Test Bench and Flight Simulator Cable Nets

The toolbox implements the functionality of test benches and flight simulators CNDS development based on AC AE device data entered into the dBricks tool. The user selects AE project which should be implemented at the test bench, enters information about the location of AC AE at the test bench, as well as information on the location of the bench equipment and technological components. Based on the data entered the package produces the following set of documents:

- a) Lists of parts (PL) required for harnesses manufacture;
- b) Harness wire lists;
- c) Tables of harnesses to technological components connections;
- d) Tables for continuity testing;
- e) CN manufacture labor cost tables (typical operations necessary for making the CN are calculated).

7.14 Toolbox for Development of Test Benches and Flight Simulators Configuration Files

The toolbox implements functionality of automatic generation of configuration files used to configure test benches and flight simulators simulation systems. List and format of files varies depending on the equipment used by the Customer, but usually includes the following main groups of files:

- a) Lists of simulated devices;
- b) Lists of simulated devices ports;
- c) Lists of input/output parameters of simulated devices functions;
- d) Lists of connections between the parameters of simulated devices functions;
- e) Description of the format of simulated ports transmitted/received data with mapping to parameters. The description depends on the simulated port type.

8 Technical Support and Tool Update Policy

The technical support of the tool consists in the listed below services provided by the Vendor. For license agreements that provide access to the tool as an Internet service the technical support is provided during the whole period of access to the tool. For license agreements that account for the installation of the tool on the Customer's local server the technical support is provided during the time specified in the license agreement and maintenance contract.

8.1 Phone Support

The support consists in providing voice consultations for Customer's representatives by the phone specified in the contract.

Unless otherwise specified in the contract, consultations are provided from 10:00 to 18:00 Moscow time on weekdays determined in accordance with the legislation of the Russian Federation.

Unless otherwise specified in the contract, the total amount of consulting should not exceed 10 hours per month.

8.2 Email Support

The support consists in providing answers to Customer questions sent to the e-mail address specified in the contract.

Unless otherwise specified in the contract, the deadline for answering questions is two business days determined in accordance with the legislation of the Russian Federation.

8.3 Response to Comments Submitted in Bug & Improvement Tracking System

Support consists in registration, processing, resolving and/or provision of a reasoned correction refusal, on the basis of Customer comments in the bug tracking system (located at the URL specified in the contract).

The response time to comments depends on the error severity. The existing error severity levels are:

- a) Blocking: the error results in total inoperability of the tool. Response period - within one calendar day.
- b) High: the error leads to a substantial decrease in the possibility of using the tool as intended. Response period - within one business day determined in accordance with the legislation of the Russian Federation.
- c) Low: the error results in minor difficulties in using the tool. Response period - within five business days determined in accordance with the legislation of the Russian Federation.
- d) Improvement Suggestion - Customer's suggestions for improving the functionality of the tool in comparison to the version used. Response period - within fifteen business days determined in accordance with the legislation of the Russian Federation.

Note: Response time is not the time to fix the problem or correct the error.

8.4 Upgrade Policy

During the technical support period the Vendor provides the Customer with information on available new versions of the tool. At the Customer's request the Vendor can upgrade the tool to the latest available version.

9 Conclusion

The dBricks system allows to:

Significantly reduce integrated systems and equipment complexes development effort,

- a) Ensure automatic control of project data changes and their reflection in project documentation,
- b) Reduce design time by increasing the speed of access for design process stakeholders to current data and its changes,
- c) Minimize the amount of equipment and optimize the architecture of the complex through fail-safety evaluation at the early stages of development,
- d) Reduce design and certification risks by automating the use of existing regulations and certification requirements.