NASA/TM—2008-215158 AIAA–2008–0443

IceVal DatAssistant—An Interactive, Automated Icing Data Management System

Laurie H. Levinson
Glenn Research Center, Cleveland, Ohio

William B. Wright
ASRC Aerospace, Cleveland, Ohio

May 2008

NASA/TM—2008-215158 AIAA–2008–0443

IceVal DatAssistant—An Interactive, Automated Icing Data Management System

Laurie H. Levinson
Glenn Research Center, Cleveland, Ohio

William B. Wright
ASRC Aerospace, Cleveland, Ohio

Prepared for the
46th Aerospace Sciences Meeting and Exhibit
sponsored by the American Institute of Aeronautics and Astronautics
Reno, Nevada, January 7–10, 2008

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

May 2008

IceVal DatAssistant—An Interactive, Automated Icing Data Management System

Laurie H. Levinson
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

William B. Wright
ASRC Aerospace
Cleveland, Ohio 44135

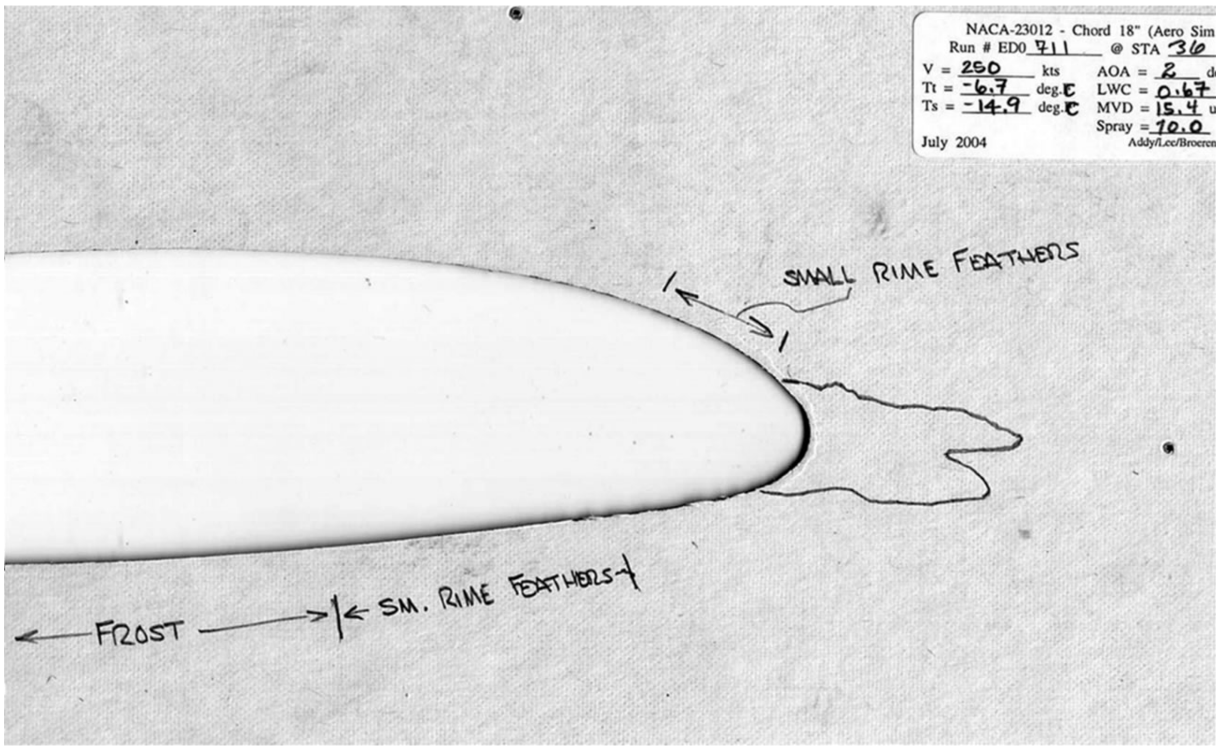
Abstract

As with any scientific endeavor, the foundation of icing research at the NASA Glenn Research Center (GRC) is the data acquired during experimental testing. In the case of the GRC Icing Branch, an important part of this data consists of ice tracings taken following tests carried out in the GRC Icing Research Tunnel (IRT), as well as the associated operational and environmental conditions documented during these tests. Over the years, the large number of experimental runs completed has served to emphasize the need for a consistent strategy for managing this data. To address the situation, the Icing Branch has recently elected to implement the IceVal DatAssistant automated data management system. With the release of this system, all publicly available IRT-generated experimental ice shapes with complete and verifiable conditions have now been compiled into one electronically-searchable database. Simulation software results for the equivalent conditions, generated using the latest version of the LEWICE ice shape prediction code, are likewise included and are linked to the corresponding experimental runs. In addition to this comprehensive database, the IceVal system also includes a graphically-oriented database access utility, which provides reliable and easy access to all data contained in the database. In this paper, the issues surrounding historical icing data management practices are discussed, as well as the anticipated benefits to be achieved as a result of migrating to the new system. A detailed description of the software system features and database content is also provided; and, finally, known issues and plans for future work are presented.

I. Introduction

As with any research group, members of the Icing Branch at the NASA Glenn Research Center (GRC) generate and use a great deal of data in the course of doing their research. For the Icing Branch, this data most often takes the form of hand-traced and digitized ice shapes (see Figs. 1 and 2), and the associated operational and environmental conditions at the time those ice shapes accrete. Over the years, the management of this data has increasingly become a major concern, as thousands of ice shapes have been generated by different researchers under varying conditions and for different purposes. In the past, the individual researchers performing a test have each been responsible for managing their own test data, meaning the location, reliability, and type of data retained has been subject to a great deal of variability. If, at a later point in time, a researcher wished to access the data, often this would require a great deal of time and effort; and, in some cases, its use would be complicated by missing or conflicting information.

To address this situation, the Icing Branch has recently chosen to implement an interactive, automated data management solution referred to as the IceVal DatAssistant software system. This system has two primary components: a relational database, used to store both experimental and simulation software-generated ice shapes and the associated conditions data; and a database access utility, used to upload, download, and/or display user-selected data. By migrating to this type of solution, members of the Icing Branch have not only been able to improve their current data management practices, saving time and effort, but they are now also able to provide an improved, more comprehensive product to their customers, while simultaneously laying the foundation for future enhancements to both products and processes.


NACA-23012 - Chord 18" (Aero Sim

Run # ED0 711 @ STA 36

V = 250 kts AOA = 2 d

Tt = -6.7 deg.C LWC = 0.67

Ts = -14.9 deg.C MVD = 15.4 u

Spray = 10.0

July 2004 Addy/Lee/Broeren

Figure 1.—Typical icing template, showing hand-traced ice shape and researcher notes documented onto the template.

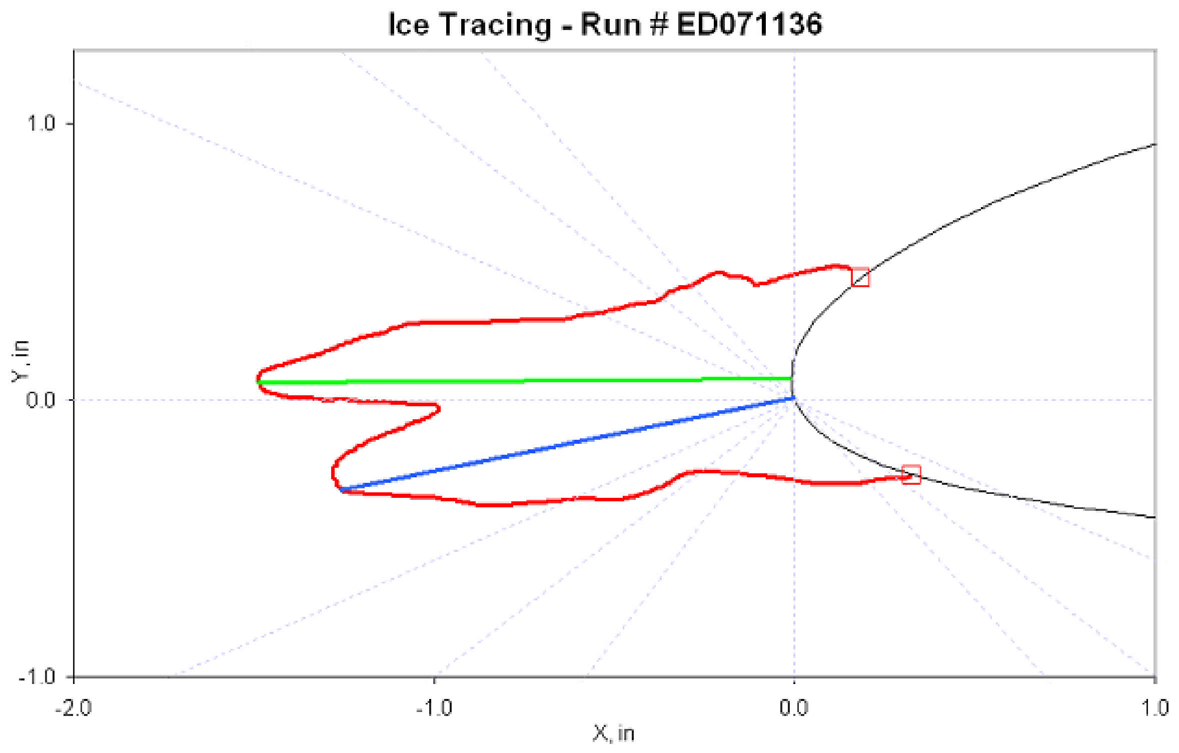


Figure 2.—Digitized version of ice shape shown above in Fig. 1. Note: Digitized ice shapes are always displayed to the left of the airfoil, whereas the orientation of a hand-traced ice shape is determined by the orientation of the airfoil in the Icing Research Tunnel. For this reason, the two versions of an ice shape may be mirror images of one another, as is the case in this example.

II. Rationale

The experimental ice shapes collected over the last 20 years by researchers at GRC have primarily been generated during tests performed in the NASA Glenn Icing Research Tunnel (IRT).[*] The IRT is an instrumented, refrigerated wind tunnel used by researchers and aircraft design engineers to simulate the natural conditions experienced by aircraft when flying through an icing cloud. A wide range of tests have been performed in the IRT over its years of operation, providing information as to how ice forms on an aircraft's surface under a given set of conditions,[1] the impact a particular ice shape has on an aircraft's performance,[2] or the effectiveness of a proposed icing prevention[3] or ice removal[4] strategy. Additionally, tests have been run in order to calibrate the tunnel after equipment repairs or upgrades,[5] to provide validation data for ice prediction software systems,[6] or for a variety of other icing research-related purposes.[†]

While the data acquired during these experimental tests has generally been collected with a specific purpose in mind, the data captured often has additional value beyond just the immediate objectives of the researcher conducting the test. For example, the results of an experiment performed for purposes of understanding how ice forms under a particular set of conditions may also be used to validate an ice shape prediction code for use under those same conditions;[7] or the results of a test conducted at one point in time may be used in planning a subsequent test, or evaluating its results.[8] In either case, having efficient, reliable access to previously collected data could save a great deal of time and effort, and provide valuable information to the research engineer. In addition, given the fact that testing in the IRT is a relatively expensive endeavor, it would certainly seem appropriate to take the necessary steps to ensure the long-term integrity and availability of any data acquired that might be of use at some later point in time.

Although the above rationale may already provide sufficient motivation for undertaking the development of the type of data management solution implemented in the IceVal DatAssistant software system, there are also a number of other reasons for the Icing Branch to migrate to this type of solution. While these reasons may differ in their specific focus, the common thread among them is the recognition that, by compiling a comprehensive database consisting of a standardized, reliable data set, and combining that with an easy-to-use graphical user interface (GUI), providing quick and easy access to the data, a number of present and future enhancements to both products and processes become possible:

1) More comprehensive, less costly product validation

In the past, validation of Icing Branch software, such as the LEWICE ice shape prediction code,[9] has been an extremely labor-intensive process. Even where automated software testing could be done to compare simulation results to measured data, collecting the necessary data and ensuring the consistency of format required to perform reliable automated comparisons has been a time-consuming, tedious, and error-prone process. Ultimately, the difficulties involved have necessitated that considerable time and resources be expended in order to complete the required work and assure a high quality product. With the availability of the IceVal system, however, a comprehensive validation data set, with a system-enforced consistent format, will already have been compiled, completely eliminating the most burdensome aspects of the software validation process, and facilitating comprehensive automated validation testing.

2) Enhanced quantitative software validation techniques

Prior to the release of LEWICE 2.0,[10] assessments as to the performance of ice shape prediction codes typically were based on subjective, qualitative criteria, such as visual comparisons of actual ice shapes measured in an experimental facility with those produced by an ice shape prediction code.[11] The use of this approach was not only due to the technical difficulties involved in performing more rigorous validation tests, as described above, but also due to the lack of predefined quantitative acceptance criteria established by the icing community. For the LEWICE 2.0 validation, however, GRC researchers compiled data from a large number of experimental icing tests (over 800 ice shapes from approximately 400 IRT test runs); and, using this data, they were able to develop an initial version of a utility called THICK. THICK is a software tool used to examine ice shapes and provide quantitative measures, such as icing limits, maximum horn thickness, horn angle, etc., which can then be used to assess the similarity of two ice shapes. Implementation of the THICK utility, and the definition of the associated quantitative measures for the evaluation of ice shapes, represented a significant step forward in the icing community's efforts to move from

[*]A detailed description of the IRT can be found on the Internet at http://facilities.grc.nasa.gov/irt/index.html.
[†]Each cited reference provides an example of the type of testing mentioned. Additional examples can be found elsewhere in the literature.

qualitative to quantitative ice prediction software validation techniques; and the first step in that process was compiling a standardized, consistently formatted data set for use both in formulating the technique and, later, in validating its implementation. With the development of the IceVal system, this data set has now been extended to an even broader range of ice shapes, airfoils, and conditions, and the format and content of the data are now rigorously controlled. Therefore, by combining this enhanced data set with the ability—through use of the GUI—to quickly view and output user-selected data subsets and associated ice shape images, the IceVal DatAssistant software system will facilitate the process of upgrading existing techniques and/or identifying new quantitative approaches.

3) Improved experimental testing

A required step in the process of developing an automated data management system is the definition of a standardized interface for data input and output. In the case of the IceVal system, the interface that has been defined is based on existing Icing Branch standards and practices for the collection of experimental data during IRT testing. By enforcing this local standards-based interface, therefore, use of the IceVal system will help to encourage a more rigorous and consistent application of the existing data collection protocol – helping to ensure that the data specified in current Icing Branch procedures is consistently captured and retained. Furthermore, with a comprehensive database of testing data consolidated in one location, and system features permitting selective display of user-specified data subsets, the system can also be used to expose existing data inconsistencies, or gaps in the collected data, providing valuable input to the experimental test planning process. For example, while planning a NACA-0012 airfoil scaling test entry, one could use the IceVal system to display the subset consisting of all scaling tests performed on the NACA-0012, in order to highlight any gaps in the existing data. Missing conditions identified in this manner could then be included in the next test entry.

4) Increased efficiency during the aircraft design process

With the release of the initial version of the IceVal system, all publicly available IRT-generated experimental ice shapes to date with complete and verifiable conditions have been compiled into one electronically-searchable database. For the aircraft design community, which must certify aircraft for flight in icing conditions, combining this electronically-searchable database with the selective display capabilities of the GUI has the potential to provide significant benefit during the design certification process. By using the IceVal system's capability to display specific user-selected test cases, as described below in detail, it will often be possible to quickly locate ice shapes from existing test cases that are associated with an airfoil that falls into the same general classification as the new airfoil to be certified. By using the ice shape from one of these existing test cases as a starting point, the design engineer will then be able to expedite the process of determining an appropriate ice shape for use during certification testing, thereby increasing the efficiency of the aircraft design certification process.

5) New and/or improved analytical methods

The foundation of all scientific inquiry is data; and, certainly, a key factor in the increased pace of scientific progress in recent times is the improved access and data processing capabilities provided by computers. In a similar manner, the consolidation of the experimental icing data into the IceVal relational database, combined with the ease of access and data processing capabilities provided by the IceVal GUI, will facilitate the process of performing statistical analyses or other scientific investigations that might ultimately lead to new and/or improved methods in the icing research field.

6) Ability to implement future enhancements to data management procedures and/or products

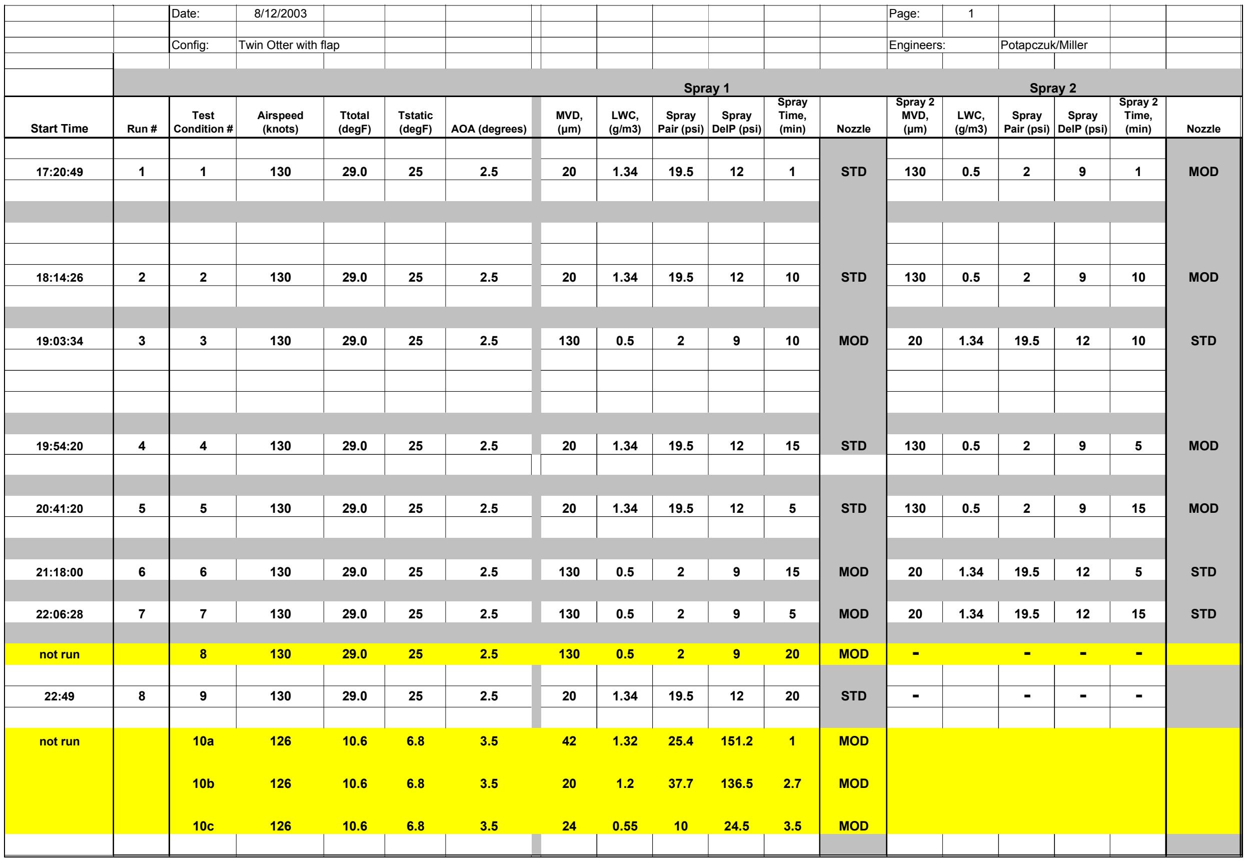
As with most organizational process improvement efforts, one of the primary benefits of implementing the IceVal system is the foundation that it will provide for future enhancements to Icing Branch data management practices and/or products. Having a system in place with a well-defined, consistent data interface will underscore for all who use the system precisely what data is and is not being routinely collected during experimental testing. This may then serve as the stimulus that will result in an upgrade to both the experimental testing protocol and the IceVal system itself, so as to capture additional, or different, data. For example, the original spreadsheets containing the experimental test data often list only the icing conditions and not the actual spray bar settings used during a test. As the equations for determining liquid water content and drop size have changed over time, the spray bar settings would be useful for resolving discrepancies in the data; but because this data is not generally documented on the spreadsheet, it is not included in the current version of the database. Likewise, the templates that the researchers use for tracing ice shapes often include notes about feather formation or anomalies witnessed during a test. While these templates are now digitally scanned, the additional information is not being transferred to the spreadsheets, and thus is also not currently included in the database. With an initial version of the system in place, however, any of these enhancements can be readily implemented in the future.

In a similar fashion, if, in the future, a new area of icing research should require that different data be captured or additional computations be routinely performed, the currently existing system can be modified to incorporate this new capability. With most efforts such as this, the most difficult step is the first one; thus, the expectation is that any future enhancements to Icing Branch data management products and processes will now require only evolutionary, not revolutionary, change.

III. Legacy Data Overview

When an effort was first made to consolidate experimental data for use in ice shape prediction code validation, the data that had been collected during IRT test runs was discovered to be in a wide variety of formats. Each researcher had been responsible for managing his or her own test data; and, although data collection and naming standards were in place at the time, the extent to which the standards were being followed, and the individual interpretations of those standards, varied. In addition, much of the documented data was stored as hardcopy information, and was thus not available electronically. The most commonly used methods for documenting data from experimental test runs were test matrices, run logs, and icing templates.

A test matrix is a chart, often in the form of an Excel spreadsheet, showing the planned or actual conditions used for an entire test entry, which generally consists of multiple test runs, or sets of conditions. An example of a test matrix is shown in Fig. 3. The run log is a handwritten, hardcopy sheet containing information from a single test run— i.e., one set of conditions—and includes information as to the actual conditions run, observations made during the test, etc. A sample run log is shown in Fig. 4, below. An icing template is a rectangular piece of cardboard with the contour of the relevant airfoil cut into it and a hand-drawn ice tracing image documented onto it. The ice tracing is created by entering the tunnel following the test and manually tracing the ice shape that formed onto the cardboard. As shown above in Fig. 1, aside from the actual ice tracing itself, the template might also contain miscellaneous notes made by the researcher during or after the test.



		Date:	8/12/2003										Page:	1				
		Config:	Twin Otter with flap										Engineers:		Potapczuk/Miller			
									Spray 1						Spray 2			
Start Time	Run #	Test Condition #	Airspeed (knots)	Ttotal (degF)	Tstatic (degF)	AOA (degrees)	MVD, (µm)	LWC, (g/m3)	Spray Pair (psi)	Spray DelP (psi)	Spray Time, (min)	Nozzle	Spray 2 MVD, (µm)	LWC, (g/m3)	Spray Pair (psi)	Spray DelP (psi)	Spray 2 Time, (min)	Nozzle
17:20:49	1	1	130	29.0	25	2.5	20	1.34	19.5	12	1	STD	130	0.5	2	9	1	MOD
18:14:26	2	2	130	29.0	25	2.5	20	1.34	19.5	12	10	STD	130	0.5	2	9	10	MOD
19:03:34	3	3	130	29.0	25	2.5	130	0.5	2	9	10	MOD	20	1.34	19.5	12	10	STD
19:54:20	4	4	130	29.0	25	2.5	20	1.34	19.5	12	15	STD	130	0.5	2	9	5	MOD
20:41:20	5	5	130	29.0	25	2.5	20	1.34	19.5	12	5	STD	130	0.5	2	9	15	MOD
21:18:00	6	6	130	29.0	25	2.5	130	0.5	2	9	15	MOD	20	1.34	19.5	12	5	STD
22:06:28	7	7	130	29.0	25	2.5	130	0.5	2	9	5	MOD	20	1.34	19.5	12	15	STD
not run		8	130	29.0	25	2.5	130	0.5	2	9	20	MOD	-		-	-	-	
22:49	8	9	130	29.0	25	2.5	20	1.34	19.5	12	20	STD	-		-	-	-	
not run		10a	126	10.6	6.8	3.5	42	1.32	25.4	151.2	1	MOD						
		10b	126	10.6	6.8	3.5	20	1.2	37.7	136.5	2.7	MOD						
		10c	126	10.6	6.8	3.5	24	0.55	10	24.5	3.5	MOD						

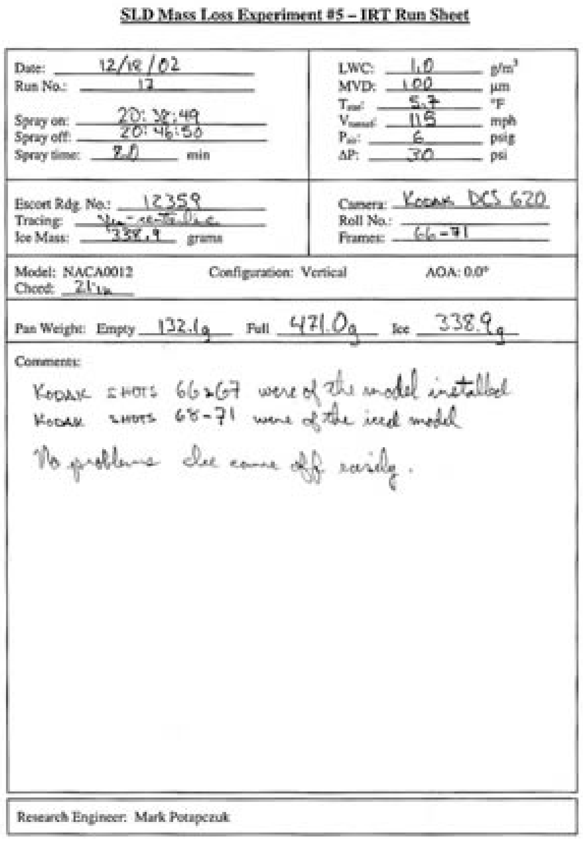
Figure 3.—Final test matrix, completed following a planned 10-run IRT test entry.

In addition to test matrices, run logs, and hand-drawn icing templates, beginning in 1991, ice tracings taken following IRT test runs were, in some cases, digitized and stored electronically (See Fig. 2, above, for an example). Routine digitization of hand-drawn ice tracings began in the mid-1990s.

Other than the above, the primary source of experimental data from IRT test runs is the data taken by the IRT's Escort-D real-time data acquisition system. This system is set up prior to the beginning of a test, and records user-selected raw data, such as fan speed, air speed, spray bar pressure, air pressure, temperature, etc., and can be used by the researchers to verify information documented on the final test matrix.

While all of this data is available, in theory, for every test run, as alluded to previously, this is often not the case in practice. Because the data is managed by the individual researcher, and much of it is stored in hardcopy form, locating a specific data set may be quite difficult. Furthermore, because of differences in experimental technique from one researcher to the next, the extent to which the relevant conditions have been explicitly documented can vary widely. Moreover, since plans made prior to a test may change during the test based on interim results, it is not uncommon to find conflicting information as to the actual conditions run, complicating interpretation of the data.

Although these types of issues are certainly not unique to the icing research environment, implementation of a special-purpose solution was nevertheless essential, so that the system's features could be tailored to the specific needs of its users. The architecture and principal features of this system, as well as the process used for its development, are described in the remaining sections of this paper.



SLD Mass Loss Experiment #5 – IRT Run Sheet

Date: 12/19/02	LWC: 1.0 g/m³
Run No.: 17	MVD: 100 µm
Spray on: 20:38:49	T_{tot}: 5.7 °F
Spray off: 20:46:50	V_{tunnel}: 115 mph
Spray time: 8.0 min	P_{air}: 6 psig
	ΔP: 30 psi
Escort Rdg. No.: 12359	Camera: Kodak DCS 620
Tracing: [illegible]	Roll No.:
Ice Mass: 338.9 grams	Frames: 66-71

Model: NACA0012 Configuration: Vertical AOA: 0.0°
Chord: 21 in

Pan Weight: Empty 132.1g Full 471.0g Ice 338.9g

Comments:
Kodak shots 66 & 67 were of the model installed
Kodak shots 68-71 were of the iced model
No problems. Ice came off easily.

Research Engineer: Mark Potapczuk

Figure 4.—Sample IRT run log.

IV. Software Development Process

The traditional software development process, utilized for many years by software practitioners, is referred to as the "waterfall model" for software development. This is a sequential process, beginning with requirements definition, and proceeding through design, implementation, verification, and maintenance in a linear fashion—completing each phase, and its associated documentation, before moving on to the next one. In recent years, this process has fallen into increasing disfavor, considered too inflexible and/or to involve too much overhead for today's fast-paced, rapidly evolving, and highly competitive market. Instead, a variety of other lifecycle models have been proposed, from the "spiral model,"[12] which has much in common with the original waterfall model, to the so-called "agile development" models,[13] such as Extreme Programming or Scrum, which have much less in common with the waterfall approach, tend to minimize documentation, and—as the name implies—emphasize rapid delivery of working software, and the ability to quickly respond to change.

For the development of the IceVal software system, the process used, best described as a "rapid" or "evolutionary prototyping"[14] approach, falls somewhere in the middle of the range in terms of formality of both process and product—still incorporating the most essential forms of documentation, although less formally implemented, and utilizing a more user-centric, iterative approach to design and implementation than typically found with the more traditional methodologies.

To begin the process, the first step was to define and document an initial set of system requirements. For the IceVal system, this was accomplished through informal discussions with users, the results of which were then documented in detail and posted to the team web site for review. Once agreement had been reached as to the initial requirements, a preliminary version of the system was implemented, with emphasis on providing a working prototype as soon as possible. Once this prototype was available, and each time a relevant upgrade in functionality was implemented, the software was demonstrated for the users; and feedback from these sessions was then integrated into both the requirements documentation and the software itself.

From a design standpoint, an effort was made to create a system that was very modular in nature, consisting primarily of small, reusable, single-function routines with relatively transparent logic. In the few cases where longer, more complex routines were required, Program Design Language (PDL), or pseudocode—a structured, English language representation of the logic—was written, and was then included as a header in the associated routine.

As the IceVal system is an interactive system, testing was primarily function-based, and was likewise accomplished in an iterative fashion; i.e., when a particular system capability was added or updated, that portion of the system was rigorously tested to ensure that all features would perform as planned. In addition, wherever possible, "round-trip" testing was also performed; i.e., a file originally used as input to the system, was compared with the associated output file generated by the system using the data originally provided by the input file.

V. Database Description

The database component of the IceVal system consists of an Access 2003 (Microsoft Corporation) database file with nine individual database tables, as described below in Table 1. The specific parameter fields defined in each table, and the assigned data type for each of those fields, are as shown in Fig. 5. In general, the data type used for a parameter field is consistent with the native type of the data, as expected. However, the exception to this is in the case where a value that would normally be expected to be read in as a number is instead read in as a text value. Where this occurs, it is either due to the fact that the input parameter associated with that field may have a non-numeric character embedded in the input field along with the parameter value itself, e.g., a 'SpanLocation' value specified as '*36"*' in the input file; or, in the alternative, the associated input parameter may not always have a numeric value at all, e.g., an 'UpperHornAngle' value may be displayed in the input file as '*N/A*.'

In terms of structure, the database tables are primarily defined along functional lines, i.e., the data is organized such that parameters are located in the same table with other parameters that have a similar purpose. Beyond this, parameters are further arranged, as needed, so as to normalize the data contained in the database, i.e., to prevent the unnecessary repetition of the same information in multiple database records. For example, the 'RunSpecs' and 'SprayConditions' tables both contain run parameters associated with the various test runs included in the database.

TABLE 1.—ICEVAL DATABASE TABLE DESCRIPTIONS

Table Name	Table Index	Table Description
AirfoilCoordinates	AirfoilName	Nominal x and y coordinates for all airfoils contained in the database
GridlineCoordinates	Degrees	Start and end x and y coordinates, used by Excel to plot image reference points displayed on the ice shape image chart
IceShapeData	RunID	Airfoil and ice shape x and y coordinates for all ice shapes contained in the database
IceThicknessData	RunID	Ice thickness vs. wrap distance values for all ice shapes contained in the database
RefConstants	Name	The set of reference constant values, and associated units, used in the calculation of run constants for each test case contained in the database
RunConstants	AltRunID	The set of run constant values associated with each test case contained in the database, calculated using data from the RunSpecs, SprayConditions, and RefConstants tables, utilizing equations found in the 2004 report by Anderson[15]
RunSpecs	RunID	Basic run parameters with values that remain unchanged, such as test date, test objective, airfoil name, etc., for each test run contained in the database
SprayConditions	AltRunID	The set of run parameters with values that may vary during an individual test run, such as liquid water content, drop diameter, and spray duration, for each test case contained in the database
ThickUtilityData	RunID	Ice shape parameter data computed by the THICK utility, such as icing limit coordinates for the upper and lower airfoil surface, upper and lower maximum ice thickness coordinates, upper and lower horn angle, etc. for each ice shape contained in the database

Field Name	Field Type
AirfoilName	Text
CoordIndex	Long Integer
AirfoilX	Double
AirfoilY	Double

a) AirfoilCoordinates

Field Name	Field Type
Degrees	Long Integer
Xstart	Single
Xend	Single
Ystart	Single
Yend	Single

b) GridlineCoordinates

Field Name	Field Type
RunID	Text
CoordIndex	Long Integer
AirfoilX	Double
AirfoilY	Double
IceShapeX	Double
IceShapeY	Double

c) IceShapeData

Field Name	Field Type
RunID	Text
CoordIndex	Long Integer
IceThickness	Double
WrapDistance	Double

d) IceThicknessData

Field Name	Field Type
Name	Text
Value	Single
Units	Text

e) RefConstants

Field Name	Field Type
AltRunID	Text
RunID	Text
LWC	Single
MVD	Single
SprayTime	Single

f) SprayConditions

Field Name	Field Type
AltRunID	Text
RunID	Text
StaticTempF	Single
StaticTempC	Single
StaticTempR	Single
StaticTempK	Single
FilmTempC	Single
FilmTempR	Single
TotalTempR	Single
MachNumber	Single
VelocitySI	Single
VelocityEng	Single
MVDFt	Single
ChordFt	Single
StaticPressure	Single
AirDensity	Single
AirViscosity	Single
SpecHeatWater	Single
AirThConductivity	Single
ChordBasedRe	Single
MVDBasedRe	Single
LEDiameterBasedRe	Single
PrandtlNum	Single
HeatXfrCoeff	Single
InertiaParamK1	Single
Lambda	Single
ModInertiaParamK0	Single
StagnationBeta	Single
ModStagnationBeta	Single
LatentHeat	Single
ScalingE	Single
HeatOfVaporization	Single
VaporPressure	Single
MassDiffusivity	Single
SchmidtNum	Single
MassXfrCoeff	Single
EvapMassLoss	Single
RelHeatFactorb	Single
DropEnergyXfrPhi	Single
AirEnergyXfrTheta	Single
FreezingPotential	Single
FreezingFraction	Single
RelHeatFactorbMod	Single
FreezingPotentialMod	Single
FreezingFractionMod	Single
FreezingFractionPerCentDiff	Single
AccumParamByChord	Single
AccumParamByLEDiameter	Single

g) RunConstants

Field Name	Field Type
RunID	Text
Date	Text
Airfoil	Text
TestObjective	Text
SpanLocation	Text
LeadEngineer	Text
Chord	Single
AirSpeed	Single
AOA	Single
CorrectedAOA	Single
TotalTemperature	Single
IPSOn	Text
SpanAngle	Text
LinkedRunID	Text
RepeatConditionType	Text
LEDiameterByChord	Single

h) RunSpecs

Field Name	Field Type
RunID	Text
IceStartIndex	Text
IceEndIndex	Text
LECylinderXcenter	Text
LECylinderYcenter	Text
AirfoilLEXstag	Text
AirfoilLEYstag	Text
IcingLimitXlow	Text
IcingLimitXhi	Text
IcingLimitYlow	Text
IcingLimitYhi	Text
IcingLimitSlow	Text
IcingLimitShi	Text
IceThicknessLowerMax	Text
IceThicknessLEMin	Text
IceThicknessUpperMax	Text
LowerIceArea	Text
UpperIceArea	Text
TotalIceArea	Text
MaxThicknessIceCoordXlower	Text
MaxThicknessIceCoordYlower	Text
MaxThicknessIceCoordXupper	Text
MaxThicknessIceCoordYupper	Text
MaxThicknessSurfCoordXlower	Text
MaxThicknessSurfCoordYlower	Text
MaxThicknessSurfCoordXUpper	Text
MaxThicknessSurfCoordYupper	Text
LowerHornAngle	Text
UpperHornAngle	Text

i) ThickUtilityData

Figure 5.—IceVal database table definitions, showing the fields included in each table and their associated data types.

However, for some of the runs in the database, several different settings are used during a single test run for the parameters contained in the 'SprayConditions' table; whereas the settings for parameters contained in the 'RunSpecs' table never change during a single run. The parameters have been separated into these two distinct tables in order to permit the storage of all values for the parameters that may have values that change, while preventing the unnecessary duplicate storage of the data that doesn't change. Records from the two tables are then linked using a common value—in this case, the 'RunID' value.

VI. Software System Functionality

The IceVal DatAssistant software system is an interactive, Windows-based application developed using Microsoft Visual Basic 6.0. As is the case for other Windows applications, from the user's perspective, operation of the system is controlled via a set of graphical forms—along with the various buttons and other graphical elements, or "controls," contained on those form—that, together, constitute the system's GUI. While this notion of the system's mechanism of operation is not entirely accurate—in reality, operation of the system is accomplished via the logic contained in the forms' "event handlers"[¶] and supporting Visual Basic modules—it nevertheless serves as a useful frame of reference for an explanation of the system's features.

The IceVal software system structure is shown below in Fig. 6. Modules whose names begin with the prefix, "frm," are the system's form modules; whereas the remaining modules represent the Visual Basic support modules. These support modules—with the exception of 'MainModule,' whose function is to initiate program execution—contain subroutines and functions utilized, either directly or indirectly, by one or more of the form modules' event handlers.

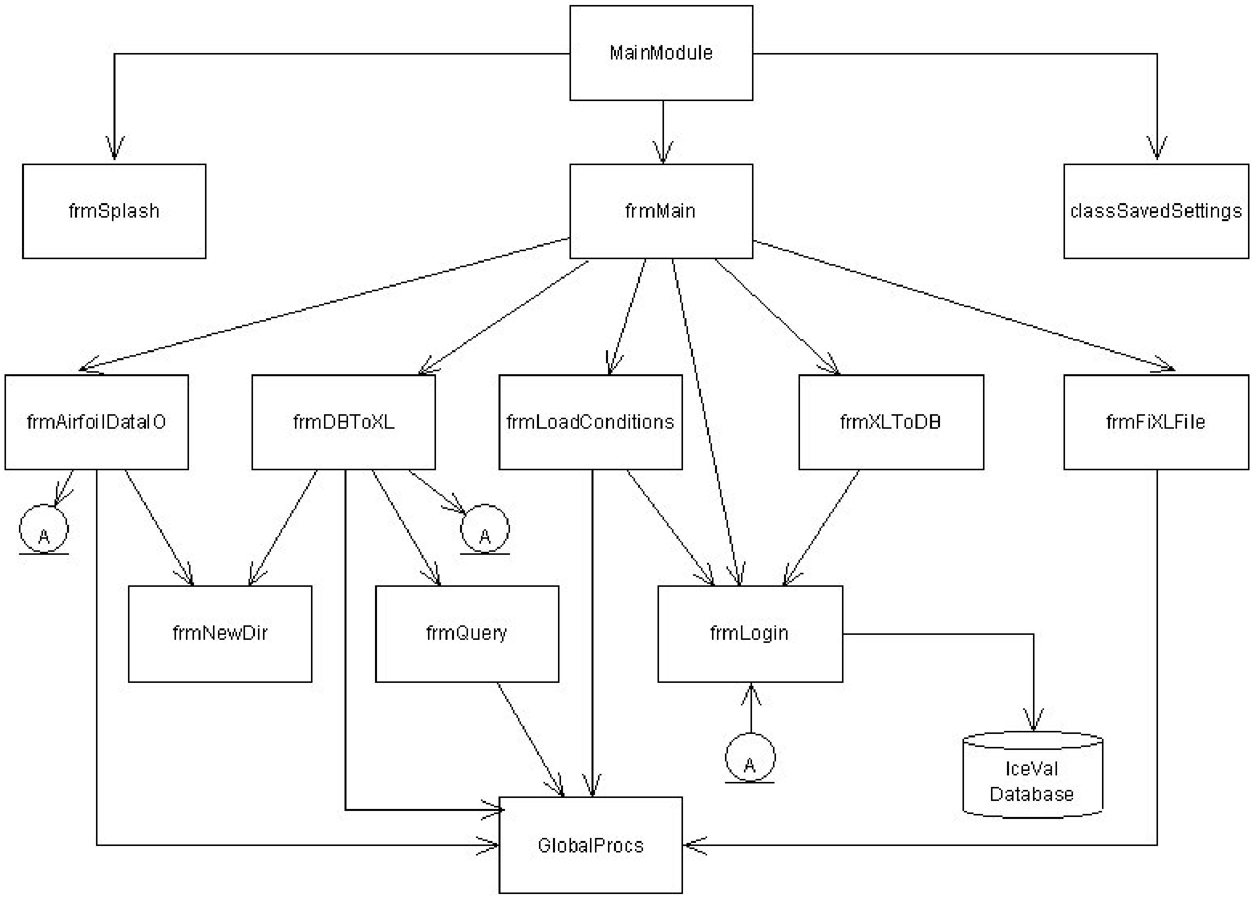


Figure 6.—IceVal DatAssistant software system structure.

[¶]An "event handler" is a subroutine that is executed in response to the occurrence of a specific "event" that occurs in the environment, such as a user clicking the mouse or entering text in a textbox.

The system's form modules, and the features implemented on each associated form, are as follows:

frmSplash

This module contains the logic for the system splash screen, shown in Fig. 7. This screen appears when the user first starts the IceVal application, and displays basic information about the software, such as the name of the application, the organization responsible for its development, and the current software version number. To close this screen, the user can either click on the form, or wait for a built-in system time-out, which closes the screen automatically after a period of 10 sec. Once the splash screen has been closed, the form for the main user screen, frmMain, is displayed immediately.

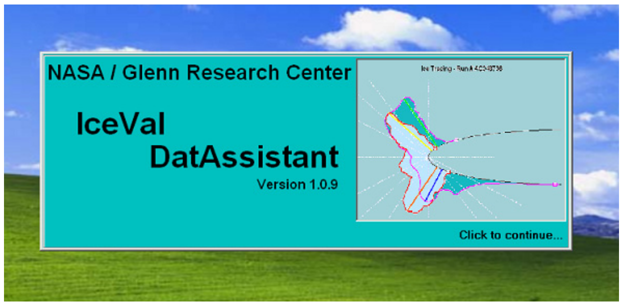


Figure 7.—System splash screen, displayed at system startup.

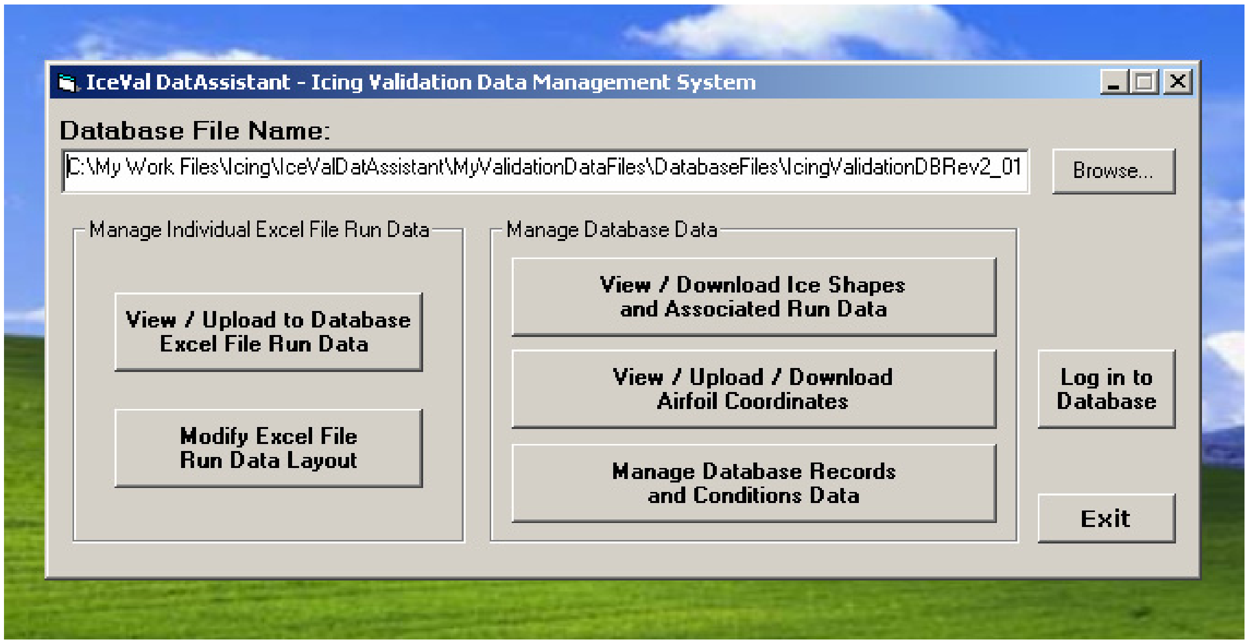
frmMain

This module contains the logic used to display and operate the main user screen. From this screen, users may select and/or log in to the database, navigate to one of several other screens defined for the application, or exit the system entirely. As shown below in Fig. 8, the form for this module consists of several CommandButton controls, each of which may be used to activate one of the other forms in the system:

The *"View/Upload to Database Excel File Run Data"* CommandButton launches the frmXLToDB.frm form module. The form for this module provides the capability to view and/or upload to the database experimental or simulation run data, including ice shape and associated run conditions data, stored in individual Excel files (see Fig. 9, below).

The *"Modify Excel File Run Data Layout"* CommandButton is used to display the form associated with the frmFiXLFile.frm module (see Fig. 10). This form provides the user with the ability to modify the layout of groups of Excel files whose data is to be uploaded to the database, but whose format does not conform to the system's expected input file format.

Selecting the *"View / Download Ice Shapes and Associated Run Data"* CommandButton activates the form associated with the frmDBToXL.frm module (see Fig. 11). This form is used to view and/or download to individual Excel files ice shape data from user-selected test runs contained in the database, along with the corresponding operational and environmental conditions. The form can also be used to save the currently displayed ice shape image to a Word file, or to copy the currently displayed image or a selected set of run IDs to the clipboard.


IceVal DatAssistant - Icing Validation Data Management System
Browse...
Manage Individual Excel File Run Data
View / Upload to Database Excel File Run Data
Modify Excel File Run Data Layout
Manage Database Data
View / Download Ice Shapes and Associated Run Data
View / Upload / Download Airfoil Coordinates
Manage Database Records and Conditions Data
Log in to Database
Exit

Database File Name: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\DatabaseFiles\IcingValidationDBRev2_01

Figure 8.—Screen shot showing the IceVal DatAssistant main user screen.

Selecting the CommandButton labeled, *"View/Upload / Download Airfoil Coordinates,"* initiates display of the form associated with the frmAirfoilDataIO.frm module (see Fig. 12). This form may be used to display and/or download to individual text files the airfoil coordinates for user-selected airfoils contained in the database. The form may also be used to upload to the database airfoil coordinates contained in user-specified text files.

Clicking on the *"Manage Database Records and Conditions Data"* CommandButton launches the form associated with the frmLoadConditions.frm module (see Fig. 13, below). This form can be used to add or delete database records, or to perform database maintenance functions, such as compacting the database or creating and initializing a new database file. It can also be used to modify conditions data values, or to view and/or output conditions data and run constants to a "master" Excel spreadsheet.

For any of the CommandButtons described above, once the user clicks on the specified CommandButton and the associated form has been activated, the frmMain form will automatically be minimized. If, however, the user subsequently either minimizes or closes that "child" form, and the frmMain form is the only other form open at the time, the frmMain form will automatically return to its normal window state, and again be displayed. Additionally, because all of the forms described above are modeless forms,[#] at any time during the display of any of these forms, the user can independently activate the minimized frmMain form and use it to open one or more of the other forms in the system.

In contrast to the situation described above, clicking on the CommandButton labeled, "Log in to Database," opens the modal form associated with the frmLogin.frm module. The default version of this dialog, used to log in to the database, is shown in Fig. 14, whereas the extended form, used to view and/or update database options, is shown in Fig. 15. Because this form is modal, until it has been closed, it is the only form in the system with which the user may interact. To close the form, the user must either provide a valid username and password and then click the "OK" CommandButton, or cancel out of the dialog by clicking the "Cancel" CommandButton or "close box" in the upper right-hand corner of the form. If the user closes the dialog without successfully logging in, however, the frmLogin form will continue to appear each time the user clicks on one of the CommandButtons contained within the frmMain Frame labeled, "Manage Database Data" (See Fig. 8, above). The user will then have another opportunity to log in to the database prior to displaying the selected form. If, at that time, the user is still unable to successfully log in to the database, the selected form will continue to be displayed, but no database data will be shown, and no functionality that requires interaction with the database will be provided.

In addition to the each of the CommandButtons described above, which are used to access other forms in the system, the frmMain form also consists of two other CommandButtons: the "Browse…" CommandButton and the "Exit" CommandButton.

The "Browse…" CommandButton is associated with the TextBox control appearing immediately to its left, which displays the name of the currently selected database file. This CommandButton invokes the standard "File Open" dialog, which can be used to navigate to the desired directory and select a database file. Once the file has been selected and the dialog closed, the associated TextBox is then updated to display the full path name for the selected file. This same combination of "Browse…" CommandButton and TextBox control displaying the database file name is also found at the top of all forms in the system from which the user may access the database, and may be used to select a database file directly from each of those forms, if desired.

The "Exit" CommandButton, located at the bottom right of the frmMain form, is used to exit the application, and when clicked, causes any of the application's open forms to be immediately closed and unloaded from memory.

frmXLToDB

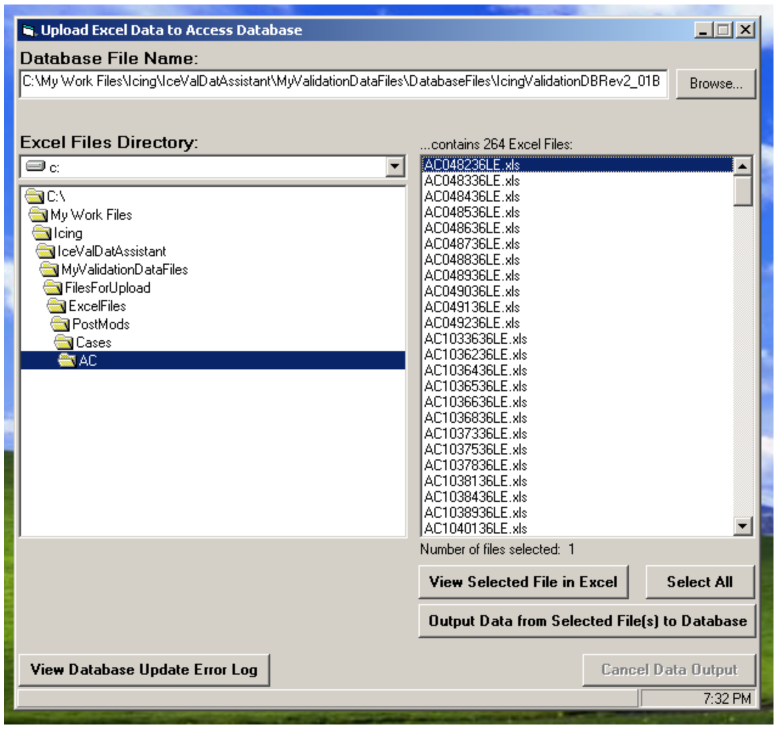
The form associated with the frmXLToDB.frm module, shown below in Fig. 9, provides the user with the capability to view and/or upload to the database ice shape and associated conditions data stored in individual Excel files.

Using the interface defined by this form, the user may navigate to a desired drive and directory, view the list of Excel files in that directory, select one or more of those files for upload to the database, and then click on the "Output Data from Selected File(s) to Database" CommandButton to initiate the upload. Once the upload has begun, for each selected file, the software will first check to see whether data already exists in the database for the specified run ID, and warn the user if so. The user will then be provided with the option of overwriting the existing data, reading in only new data, or aborting the upload entirely. (The upload can also be aborted at any time by clicking on the CommandButton labeled, "Cancel Data Output," located in the lower right corner of the form, and enabled as

[#]A "modeless" form is a form that allows the user to interact with the application's other open forms while it is also open. A "modal" form does not permit the user to interact with any other open window in the application until the modal form is hidden or unloaded from memory.

soon as the upload process has begun.) Assuming the user chooses to continue the upload, or the data being read in is new data, the software will then update the database with the data contained in each selected Excel input file. In addition, for each test case processed, the set of run constants listed above in Fig. 5(g) will also be calculated and stored in the database.

Throughout the process, any error that occurs will trigger the display of a message box containing information about the error. In most cases, that same information will also be logged to an error message file created, and uniquely named, each time data is uploaded to the database. For most errors, when an error message is output to the screen, the user is also given the option to cancel all further output of similar error messages for the duration of the upload. Doing so only affects output of error information to the screen, however, and has no effect on error message information being logged to the error message file. Canceling error message output in this manner, therefore, provides the user with the ability to prevent the repeated interruption of the upload to report the occurrence of the same type of error, while still permitting comprehensive error information to be captured for later display. Once the upload has completed, the user can then choose to display the complete list of errors that occurred by clicking on the "View Database Update Error Log" CommandButton, and displaying the associated error log file.


Upload Excel Data to Access Database

Database File Name: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\DatabaseFiles\IcingValidationDBRev2_01B

Excel Files Directory: c:

C:\
My Work Files
Icing
IceValDatAssistant
MyValidationDataFiles
FilesForUpload
ExcelFiles
PostMods
Cases
AC

...contains 264 Excel Files:

AC048236LE.xls
AC048336LE.xls
AC048436LE.xls
AC048536LE.xls
AC048636LE.xls
AC048736LE.xls
AC048836LE.xls
AC048936LE.xls
AC049036LE.xls
AC049136LE.xls
AC049236LE.xls
AC1033636LE.xls
AC1036236LE.xls
AC1036436LE.xls
AC1036536LE.xls
AC1036636LE.xls
AC1036836LE.xls
AC1037336LE.xls
AC1037536LE.xls
AC1037836LE.xls
AC1038136LE.xls
AC1038436LE.xls
AC1038936LE.xls
AC1040136LE.xls

Number of files selected: 1

7:32 PM
Browse...
View Selected File in Excel
Select All
Output Data from Selected File(s) to Database
View Database Update Error Log
Cancel Data Output

Figure 9.—The frmXLToDB form, used to view and/or upload test run data from individual Excel input files.

In addition to providing the user with the ability to upload Excel file data to the database, as described above, the frmXLToDB form may also be used to locate and view a file in Excel. To accomplish this, the user must first use the provided interface to navigate to the desired drive and directory. Then, after locating the relevant file name among the list of files being displayed, the user may either double-click on that file name, or highlight the name and click on the "View Selected File in Excel" CommandButton, to open the file in a separate Excel file window.

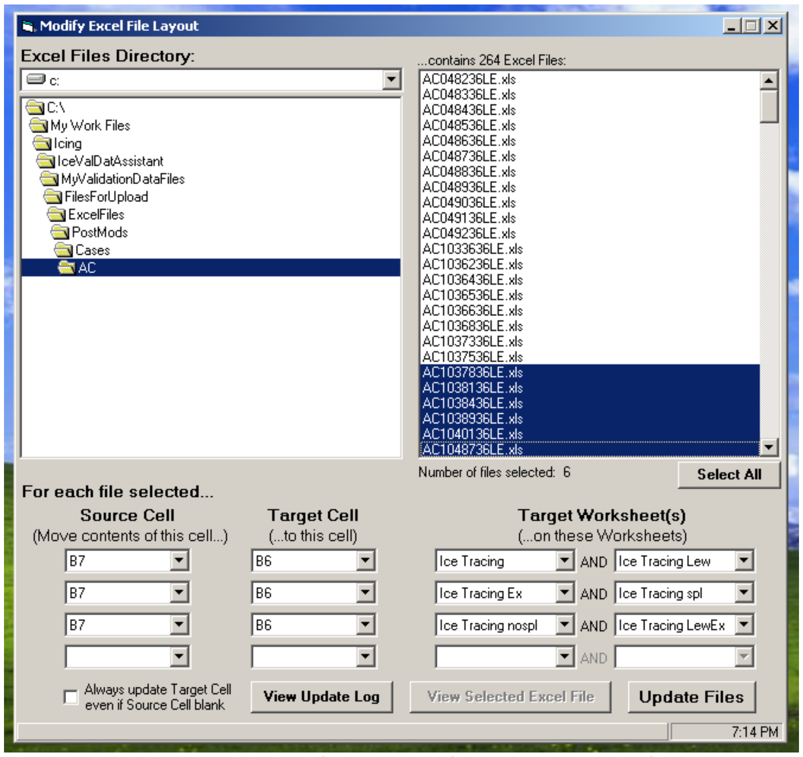
frmFiXLFile

The form associated with the frmFiXLFile.frm module, displayed in Fig. 10, is provided for the purpose of correcting the format of Excel input files whose layout does not conform to the system's expected input file format. While the form may be used to specify from one to four changes to be made simultaneously on up to two worksheets in each of one or more Excel files, use of the form is most beneficial when the user has many files all of which require the same set of changes.

As with the frmXLToDB form described above, the frmFiXLFile form should first be used to navigate to the appropriate drive and directory. Once there, if the user is prepared to proceed immediately with the update, the files to be updated can be selected. If, however, the user is unsure of exactly what changes are required, or would like to confirm the validity of an anticipated set of changes, the form may first be used to review a selected file's layout in Excel. To accomplish this, after navigating to the relevant directory, the user may either double-click one of the file names listed, or highlight a file name and click the CommandButton labeled "View Selected Excel File," and the file will open in a separate Excel window.

Once any necessary file review has been completed, and the user is prepared to proceed with the update, the desired change(s) must then be specified on the form. To do so, the user must identify, for each change, the cell location for the value which is to be moved (the "Source Cell"), the cell to which that value is to be moved (the "Target Cell"), and the name of either one or two worksheets for which the specified move is being requested (the "Target Worksheet(s)"). As shown in Fig. 10, below, ComboBoxes with dropdown lists are provided on the form for each of these required selections, with a single row to be used to specify each requested update. In addition, the user must also specify whether to proceed with a requested change even if the specified Source Cell is empty. This is accomplished by checking or unchecking the CheckBox found in the lower left corner of the form. By default, the box is unchecked, indicating that the user does not wish to perform a requested update if the Source Cell is empty. However, if, for example, the user wishes to use this form to clear the contents of a particular cell in one or more Excel input files, checking the CheckBox will enable that option.

During the change specification process, all requested changes are checked for validity, and illegal combinations—such as specifying the same cell as both the Source and Target Cell—are immediately flagged. A warning message is displayed to the user identifying the source of the error, and the "Update Files" CommandButton is disabled until the problem has been corrected.


Modify Excel File Layout

Excel Files Directory:

c:

C:\
My Work Files
Icing
IceValDatAssistant
MyValidationDataFiles
FilesForUpload
ExcelFiles
PostMods
Cases
AC

...contains 264 Excel Files:

AC048236LE.xls
AC048336LE.xls
AC048436LE.xls
AC048536LE.xls
AC048636LE.xls
AC048736LE.xls
AC048836LE.xls
AC048936LE.xls
AC049036LE.xls
AC049136LE.xls
AC049236LE.xls
AC1033636LE.xls
AC1036236LE.xls
AC1036436LE.xls
AC1036536LE.xls
AC1036636LE.xls
AC1036836LE.xls
AC1037336LE.xls
AC1037536LE.xls
AC1037836LE.xls
AC1038136LE.xls
AC1038436LE.xls
AC1038936LE.xls
AC1040136LE.xls
AC1048736LE.xls

Number of files selected: 6

For each file selected...

Source Cell (Move contents of this cell...)	Target Cell (...to this cell)	Target Worksheet(s) (...on these Worksheets)		
B7	B6	Ice Tracing	AND	Ice Tracing Lew
B7	B6	Ice Tracing Ex	AND	Ice Tracing spl
B7	B6	Ice Tracing nospl	AND	Ice Tracing LewEx
			AND	

☐ Always update Target Cell even if Source Cell blank

7:14 PM

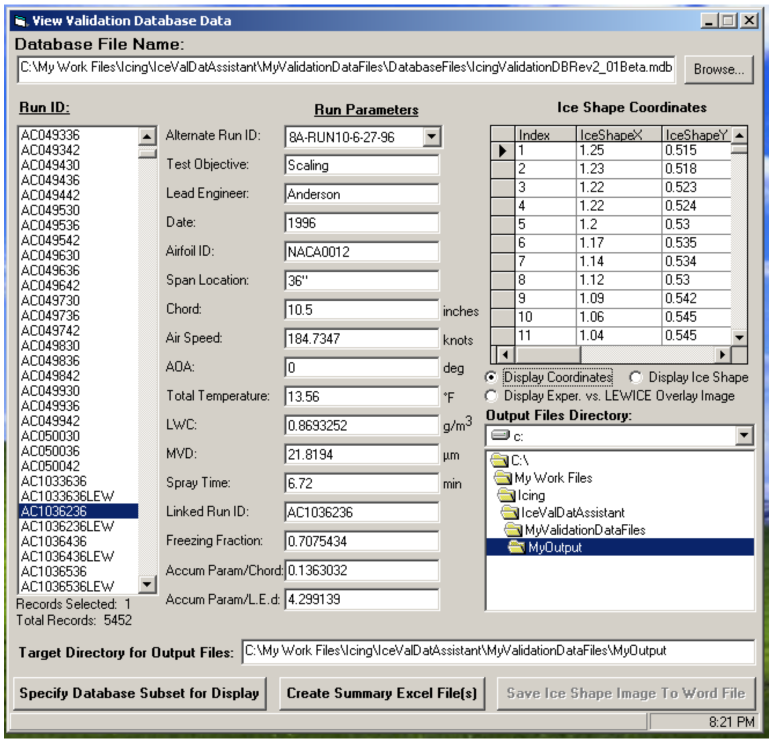
Figure 10.—The frmFiXLFile form, used to modify the layout of input Excel files.

Once all relevant selections have been made and have passed the system validity checks, the "Update Files" CommandButton is enabled, and the file update process may proceed. As the user may specify up to four distinct changes on each of two worksheets at one time, up to eight cell moves may be completed with one update command. Furthermore, because the cell update process is implemented as a simultaneous update for all moves requested during the same update, two cells' contents may be "swapped" using this form; e.g., if, on the first line of the form, a user specifies that the contents of cell A1 is to be moved to cell B6 on a particular worksheet, and on the next line, specifies that the contents of cell B6 is to be moved to cell A1 on the same worksheet, if the user then clicks the "Update Files" CommandButton, the effect will be to swap the contents of cells A1 and B6 on the specified worksheet.

While the purpose of this form is, nominally, to enable the user to specify a set of cell moves that are to be performed for a selected set of files, a key feature provided by the form is, in fact, its ability to distinguish when not to perform a requested update. During the update process, prior to completing each requested update for a selected file, consistency checks are performed to ensure that the requested update, if made, would result in a file which is compatible with the standard defined for Excel input files. If, for a given file, it is determined that this is not the case, the selected value will not be moved, and the requested update will be ignored. This allows the user to select a group of files and request a specific change without knowing, in every case, if every file selected actually requires the requested change. For example, if the user knows that several, but not necessarily all, Excel files in a particular directory have the test date in the wrong location, the move required to correct the situation could be specified on one line of this form. The user could then select all files in the directory for update, knowing that any update that would result in moving something other than a date into the specified Target Cell will be ignored. This allows the user to make the required change in all applicable files without needing to know specifically which files actually require the change. Since detailed information about each requested move is output to an update log during the update process, the user can view a summary of the results following completion of the update. By clicking on the "View Update Log" CommandButton and reviewing the associated log file, the user can thus verify that the desired results were obtained without having to examine each individual Excel file selected for update.

frmDBToXL

The form rendered by the frmDBToXL.frm module, shown in Fig. 11, below, enables the user to view a subset of the data associated with each test run contained in the database and, if desired, to output data from one of more of those test runs to individual files or to the system clipboard.



View Validation Database Data

Database File Name: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\DatabaseFiles\IcingValidationDBRev2_01Beta.mdb [Browse...]

Run ID:

AC049336
AC049342
AC049430
AC049436
AC049442
AC049530
AC049536
AC049542
AC049630
AC049636
AC049642
AC049730
AC049736
AC049742
AC049830
AC049836
AC049842
AC049930
AC049936
AC049942
AC050030
AC050036
AC050042
AC1033636
AC1033636LEW
AC1036236
AC1036236LEW
AC1036436
AC1036436LEW
AC1036536
AC1036536LEW

Records Selected: 1
Total Records: 5452

Run Parameters

Alternate Run ID:	8A-RUN10-6-27-96	
Test Objective:	Scaling	
Lead Engineer:	Anderson	
Date:	1996	
Airfoil ID:	NACA0012	
Span Location:	36"	
Chord:	10.5	inches
Air Speed:	184.7347	knots
AOA:	0	deg
Total Temperature:	13.56	°F
LWC:	0.8693252	g/m^3
MVD:	21.8194	µm
Spray Time:	6.72	min
Linked Run ID:	AC1036236	
Freezing Fraction:	0.7075434	
Accum Param/Chord:	0.1363032	
Accum Param/L.E.d:	4.299139	

Ice Shape Coordinates

Index	IceShapeX	IceShapeY
1	1.25	0.515
2	1.23	0.518
3	1.22	0.523
4	1.22	0.524
5	1.2	0.53
6	1.17	0.535
7	1.14	0.534
8	1.12	0.53
9	1.09	0.542
10	1.06	0.545
11	1.04	0.545

[✓ Display Coordinates] Display Ice Shape
Display Exper. vs. LEWICE Overlay Image

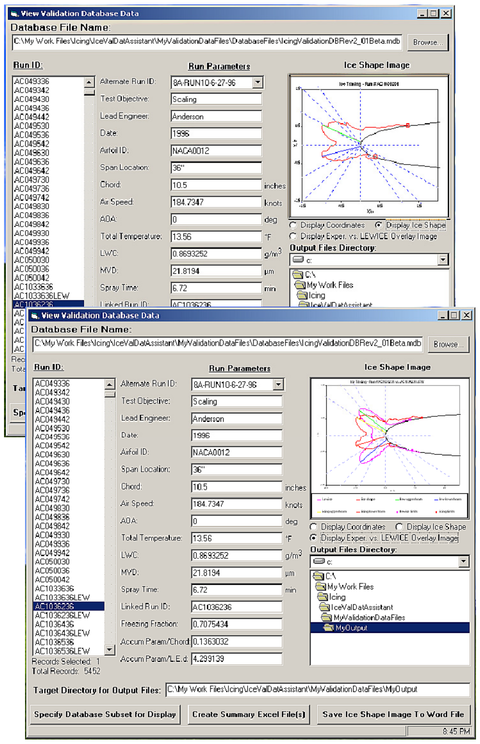
Output Files Directory: c:
C:\
My Work Files
Icing
IceValDatAssistant
MyValidationDataFiles
MyOutput

Target Directory for Output Files: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\MyOutput

[Specify Database Subset for Display] [Create Summary Excel File(s)] [Save Ice Shape Image To Word File]

8:21 PM

(a) Default version of screen, with scrollable ice shape coordinates displayed at upper right. The user may choose to display the associated ice shape or overlay image in place of the coordinates.



View Validation Database Data

Database File Name: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\DatabaseFiles\IcingValidationDBRev2_01Beta.mdb [Browse]

Run ID: | Run Parameters | Ice Shape Image

Display Coordinates [✓ Display Ice Shape]
Display Exper. vs. LEWICE Overlay Image

View Validation Database Data

Display Coordinates Display Ice Shape
[✓ Display Exper. vs. LEWICE Overlay Image]

Target Directory for Output Files: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\MyOutput

[Specify Database Subset for Display] [Create Summary Excel File(s)] [Save Ice Shape Image To Word File]

8:45 PM

(b) Alternative screens available to the user, showing the ice shape image (top) and overlay image (bottom).

Figure 11.—The frmDBToXL form, used to view and/or output user-selected database data.

To view the data associated with a particular test run, the user must first select the relevant run ID from the alphabetically-arranged, scrollable list contained in the form's ListBox control, displayed along the left-hand side of the screen. Selection of an ID may be accomplished either by scrolling to the applicable portion of the list, or by typing in the first few characters of the relevant ID, and then clicking on the desired ID. Alternatively, the keypad keys may also be used to move up or down the list via the keyboard. Regardless of the method used, each time a new run ID is selected, the operational and environmental conditions associated with the selected test run, as well as either the ice shape coordinates, ice shape image, or experimental versus LEWICE-generated ice shape overlay image resulting from that run will be displayed on the screen. The choice as to which form of the ice shape data to display is made by selecting the appropriately labeled OptionButton control,[‡] located immediately below the portion of the screen reserved for the display of ice shape data.

While most test runs in the database have only one set of operational and environmental conditions associated with the run, in some cases, a subset of the parameters might vary during a single test (See Table 5(f) above, for a list of the parameters that may vary). This is most often the case for those tests intended to simulate supercooled large droplet (SLD) icing conditions, which are typically characterized by a bimodal droplet distribution. Since it is not possible to reproduce this bimodal distribution directly in the IRT, the required droplet distribution is modeled instead by alternating between the relevant conditions over the course of the test. For any such test whose data is included in the database, a unique "AltRunID" identifier is assigned to each segment of the test run, and the "Alternate Run ID" ComboBox displayed on the screen provides the user with the means to view the associated sets of conditions. By clicking on the ComboBox dropdown arrow and selecting the appropriate identifier from the list provided, the user can display the parameter values associated with a particular segment of the currently selected multi-part run.

In addition to displaying database data for the user to view, the frmDBToXL form may also be used to output data, either to individual files or to the system clipboard.

The "Create Summary Excel File(s)" CommandButton, found in the center of the form, near the bottom, is used to output ice shape and associated conditions data from one or more user-selected run IDs to individual Excel files.

[‡]OptionButton controls are sometimes also referred to as "radio buttons."

The files created are formatted in the same manner as the files originally used to upload data to the database; and, as each file is created, a ProgressBar along the bottom of the screen updates to show progress made in writing out the specified data.

Likewise, if an ice shape image associated with an individually selected run ID is currently being displayed, the “Save Ice Shape Image To Word File” CommandButton will also be enabled, and may be used to output the displayed image to a Microsoft Word file.

In either of the above cases, any output file created is named according to the run ID associated with the data being output, and is saved to the directory specified in the read-only TextBox, found near the bottom of the form. The specified output directory is controlled, and may be updated by the user, via the DriveListBox and DirListBox controls found on the right side of the form, immediately above the associated TextBox. When selecting an appropriate output files directory, the user may either choose an existing directory, or create a new one. Caution should be exercised, however, whenever writing data to an existing directory, as any naming conflict that occurs during data output will result in the existing file being overwritten without warning.

In the event that a new output files directory is to be created, the user must first use the DriveListBox and DirListBox controls to navigate to the directory in which the new subdirectory is to be created, then right-click on the DirListBox control. Doing so will activate the frmNewDir form, shown below in Fig. 16. This form, used by both the frmDBToXL and frmAirfoilDataIO forms to create a new subdirectory, is described in detail below.

In addition to providing the ability to output data to individual Microsoft Excel and Word files, the frmDBToXL form may also be used to copy the currently displayed ice shape image, a group of selected run IDs, or the text contained in any one of the form’s TextBoxes to the system clipboard.

To copy an ice shape image, the user must first right-click on the image being displayed, then select the “Copy Image to Clipboard” Popup menu item that appears on the screen. The current image will then be copied onto the clipboard, and the user may subsequently paste that image into any application that accepts a bitmap image.

To copy a set of selected run IDs, the user must first highlight in the usual manner any run IDs that are to be copied. Alternatively, if all IDs in the ListBox are to be copied, the user has the option to right-click on the ListBox and select the “Select All” menu item from the Popup menu that appears. Once the appropriate IDs have been selected, the user may then right-click on the ListBox, and select the “Copy Selected Text” menu item from the Popup menu. This will copy a list of all selected IDs onto the clipboard, and the data may then be pasted into any application that accepts text input.

To copy text from one of the form’s TextBoxes, the user must first highlight the relevant text, then right-click the TextBox and select “Copy.” This is likely to be most useful in conjunction with the frmQuery form (see Fig. 17), activated by clicking on the “Specify Database Subset for Display” CommandButton, found in the bottom left-hand corner of the form. The frmQuery form, to be described in greater detail below, is used to select a specific subset of database records for display. Therefore, the user may wish to copy a value from one of the frmDBToXL TextBoxes, and use that value on the frmQuery form to specify an associated database subset.

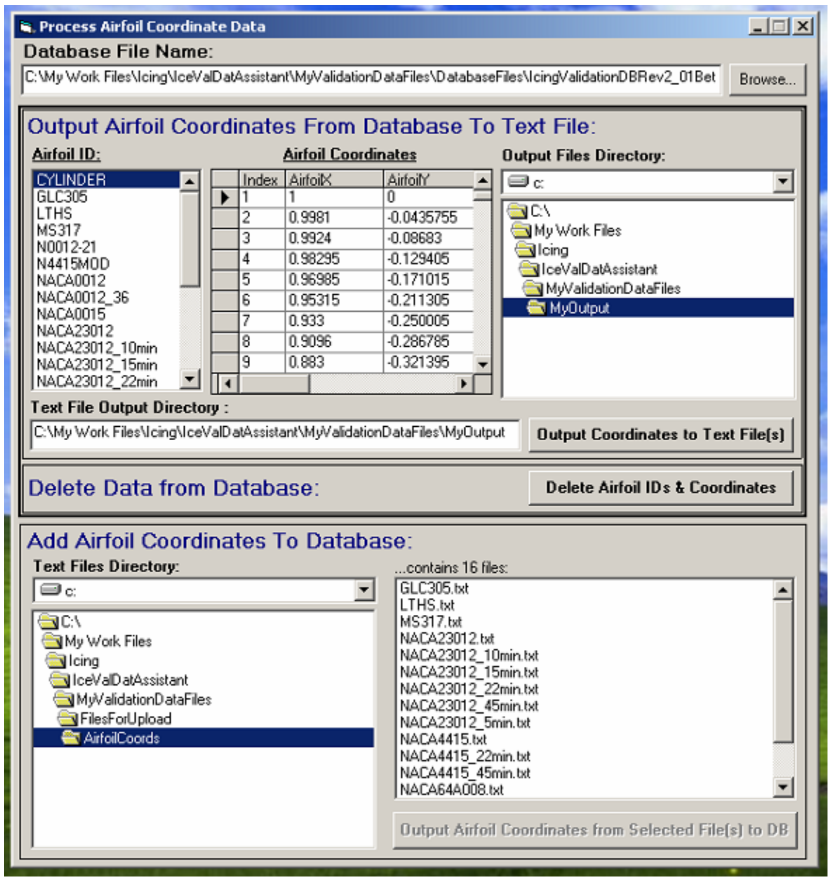
Given that the purpose of the frmDBToXL form is to display and/or output data contained in the database, central to its function is the ability to display at all times the current database data. When the form is first activated by the user, by definition, this is the case. However, because the form is modeless, as described above, other forms in the system may also be accessed while this form is open. This provides an opportunity for data inconsistencies to occur, as data contained in the database may be updated by the user from a different form after this form has been opened and initialized. For this reason, the frmDBToXL form includes a “refresh” capability, invoked by right-clicking on the run ID ListBox, and selecting “Refresh” from the Popup menu. By performing a refresh, the user can ensure that the data being displayed on the form is the current data contained in the database.

frmAirfoilDataIO

The form associated with the frmAirfoilDataIO.frm module (Fig. 12) is used to download, display and/or delete user-selected airfoil coordinates from the database, or to upload new or revised coordinates to the database. The form contains many of the same features as described above for other forms in the system, with the primary distinction being that this form deals exclusively with airfoil coordinate data.

Because the form is used for both data input and output, the display is divided into two main sections. The top half implements the functionality that requires access to existing database data, while the bottom half provides the interface needed to perform database updates, either by adding to or modifying previously loaded data. The top half of the form is further divided into two subsections, with one (the “output” section) being used to view and/or output to text files selected database data, and the other (the “delete” section) used to delete records from the database.

The frmAirfoilDataIO “output” section is similar in function to the frmDBToXL form, displaying airfoil coordinate data instead of ice growth test run data, but having many of the same types of controls and features. At


Process Airfoil Coordinate Data
Browse...
Output Coordinates to Text File(s)
Delete Airfoil IDs & Coordinates
Output Airfoil Coordinates from Selected File(s) to DB

Database File Name: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\DatabaseFiles\IcingValidationDBRev2_01Bet

Output Airfoil Coordinates From Database To Text File:

Airfoil ID:
CYLINDER
GLC305
LTHS
MS317
N0012-21
N4415MOD
NACA0012
NACA0012_36
NACA0015
NACA23012
NACA23012_10min
NACA23012_15min
NACA23012_22min

Airfoil Coordinates

Index	AirfoilX	AirfoilY
1	1	0
2	0.9981	-0.0435755
3	0.9924	-0.08683
4	0.98295	-0.129405
5	0.96985	-0.171015
6	0.95315	-0.211305
7	0.933	-0.250005
8	0.9096	-0.286785
9	0.883	-0.321395

Output Files Directory: c:
C:\
My Work Files
Icing
IceValDatAssistant
MyValidationDataFiles
MyOutput

Text File Output Directory : C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\MyOutput

Delete Data from Database:

Add Airfoil Coordinates To Database:

Text Files Directory: c:
C:\
My Work Files
Icing
IceValDatAssistant
MyValidationDataFiles
FilesForUpload
AirfoilCoords

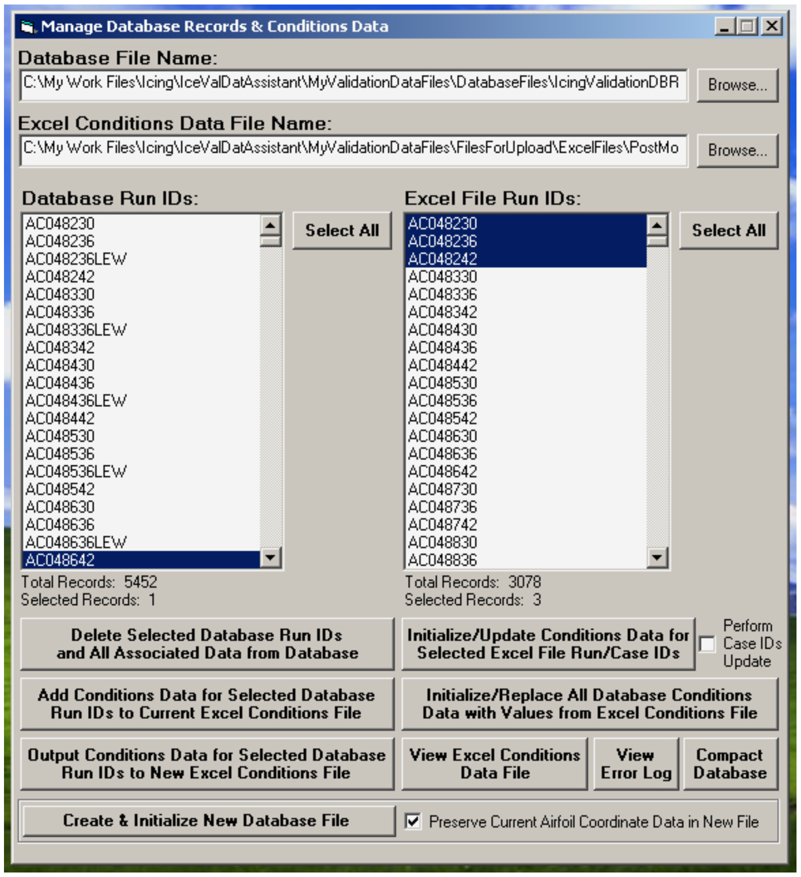
...contains 16 files:
GLC305.txt
LTHS.txt
MS317.txt
NACA23012.txt
NACA23012_10min.txt
NACA23012_15min.txt
NACA23012_22min.txt
NACA23012_45min.txt
NACA23012_5min.txt
NACA4415.txt
NACA4415_22min.txt
NACA4415_45min.txt
NACA64A008.txt

Figure 12.—The frmAirfoilDataIO form, used to process airfoil coordinate data.

the top left is a ListBox control containing an alphabetically-arranged list of airfoil IDs. To display the coordinates associated with a particular airfoil, the relevant ID is selected from the list, and the corresponding coordinates are displayed in the scrollable DataGrid located in the center of the form. To output airfoil coordinates from the database to one or more text files, the appropriate IDs are first selected from the ListBox, and the "Output Coordinates to Text File(s)" CommandButton clicked. The coordinates are then output to individual files, named according to the airfoil ID. The files are created in the directory whose name is displayed in the read-only TextBox located in the center of the form, immediately below the DataGrid control used to display the coordinates. As was the case for the frmDBToXL form, the directory name specified in the TextBox is controlled via the DriveListBox and DirListBox controls, displayed at top right; and the user may select an existing directory, or may create a new one by right-clicking on the DirListBox to activate the frmNewDir form. As before, however, caution must be exercised if the user should decide to output data to an existing directory. If that directory already contains coordinate data files, any naming conflicts that arise will result in the existing files being overwritten without warning.

Closely associated with, and immediately below, the "output" portion of the screen is the "delete" section, used to remove airfoil coordinate data from the database. The CommandButton contained in this section, labeled "Delete Airfoil IDs & Coordinates," is used to initiate the record removal process, and is enabled by selecting at least one airfoil ID from the form's Airfoil ID ListBox. Once enabled, the user may click on the CommandButton to delete from the database all coordinates associated with the selected airfoil ID(s). To protect against accidental deletion, prior to removing records from the database, a dialog is displayed to warn the user that airfoil coordinates are about to be deleted. The dialog presents the user with the option of either continuing with the deletion or aborting the operation without deleting any data. Should the user choose to continue with the deletion, all selected airfoil coordinates will be permanently deleted without further warning, and the Airfoil ID ListBox display will be updated accordingly. Although, in most cases, this screen update will proceed without delay, to enhance the robustness of the software, a time-out of 1 min is included for the update to complete. If the display should fail to update within this timeframe, the user has the option to then right-click on the ListBox to invoke the "Refresh" Popup menu. As in the case of the frmDBToXL form, this refresh capability provides the user with the means to ensure that the most recent database data is, at all times, being displayed on the screen.

The final section—and bottom half—of the frmAirfoilDataIO form is the portion of the display used to upload new or revised airfoil coordinate data from individual text files to the database. To perform this task, the user must first navigate to the relevant directory utilizing the DriveListBox and DirListBox controls found on the lower left, then select any files with data to be uploaded from the FileListBox control, located immediately to the right. Alternatively, if the user wishes to review the contents of a file before uploading data to the database, this may be accomplished by double-clicking the applicable file name prior to selecting files for upload. The specified file will then open in a separate window, and the user may view and/or modify the file's content, as necessary. When ready to proceed, the relevant files may then be selected, and the upload initiated by clicking on the "Output Airfoil Coordinates from Selected File(s) to DB" CommandButton. For each selected file, the software will then create an airfoil ID based on the name of the file containing the coordinates, and check to see whether the database already contains data for the specified ID. If data is found to exist, a warning dialog will be displayed providing the user with the option of either overwriting the existing data, skipping the update for that ID only, or canceling the upload completely. If the user chooses to overwrite the existing data, or if the database does not yet contain data for the current ID, the software will read the coordinate data from the selected file into the database, and then repeat the entire process for the next selected file. If, however, at any point, an error should occur, an error message will be displayed to the user, and all remaining updates will be aborted. At the completion of the update process, a message will then be displayed to the user indicating the total number of updates requested, and the number of updates successfully completed.



Manage Database Records & Conditions Data
Database File Name:
Browse...
Excel Conditions Data File Name:
Browse...
Database Run IDs:
Select All
Excel File Run IDs:
Select All
Delete Selected Database Run IDs and All Associated Data from Database
Initialize/Update Conditions Data for Selected Excel File Run/Case IDs
Perform Case IDs Update
Add Conditions Data for Selected Database Run IDs to Current Excel Conditions File
Initialize/Replace All Database Conditions Data with Values from Excel Conditions File
Output Conditions Data for Selected Database Run IDs to New Excel Conditions File
View Excel Conditions Data File
View Error Log
Compact Database
Create & Initialize New Database File
Preserve Current Airfoil Coordinate Data in New File

Database File Name: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\DatabaseFiles\IcingValidationDBR

Excel Conditions Data File Name: C:\My Work Files\Icing\IceValDatAssistant\MyValidationDataFiles\FilesForUpload\ExcelFiles\PostMo

Database Run IDs: AC048230, AC048236, AC048236LEW, AC048242, AC048330, AC048336, AC048336LEW, AC048342, AC048430, AC048436, AC048436LEW, AC048442, AC048530, AC048536, AC048536LEW, AC048542, AC048630, AC048636, AC048636LEW, AC048642

Total Records: 5452
Selected Records: 1

Excel File Run IDs: AC048230, AC048236, AC048242, AC048330, AC048336, AC048342, AC048430, AC048436, AC048442, AC048530, AC048536, AC048542, AC048630, AC048636, AC048642, AC048730, AC048736, AC048742, AC048830, AC048836

Total Records: 3078
Selected Records: 3

Perform Case IDs Update [ ]
Preserve Current Airfoil Coordinate Data in New File [✓]

Figure 13.—The frmLoadConditions form, used to manage the database, and to upload or output database conditions data.

frmLoadConditions

The form associated with the frmLoadConditions.frm module (Fig. 13) implements a variety of database and/or records management functions, ranging from compacting the selected database, to downloading or making changes to current database data, to creating and initializing an entirely new database file. While the features provided are not likely to be utilized with the same frequency as the features provided on some of the system's other forms, the functionality implemented greatly enhances system flexibility and extensibility, and also offers a means to obtain a comprehensive summary of data contained in the database.

The primary features implemented on this form, listed below in Table 2, may be classified as belonging to one of three categories: database data updates, conditions data downloading and/or viewing, and system maintenance operations. Within each of these categories are three individual features, implemented via CommandButtons found in one of four rows along the bottom of the screen. For most of the form's database input/output features, prior to clicking on the relevant CommandButton, it is first necessary to enable the button by selecting one or more run IDs from either the Database Run IDs or Excel File Run IDs ListBox (see Table 2). If all IDs in the ListBox are to be selected, the "Select All" CommandButton, located to the right of the ListBox, may be used to select all IDs with one button click. In addition, as noted in Table 2, for any function which updates the database in any way - including compacting the current database or creating and initializing a new database file - the user must be logged in to the database with "write access" in order to successfully execute the command. For commands involving either database input or output, once command execution has begun, a ProgressBar will be displayed along the bottom of the form, allowing the user to track the completion status of the requested operation. If, for any reason, the operation should fail to complete successfully, a dialog box will be displayed containing information as to the cause of the failure.

In addition to the features described below in Table 2, the form also includes a "Refresh" capability, similar to that provided for other forms in the system. With this feature, if data being displayed on the form is modified elsewhere—either from another form in the system, or from within Excel – the user is able to update the display

TABLE 2.—COMMANDBUTTON-ACCESSIBLE FEATURES ON THE FRMLOADCONDITIONS FORM

Feature Category	**CommmandButton Caption**	**Feature Description**	**Database / Excel File Run ID Selection Required?**	**Access Level Required**
Database Data Updates	Delete Selected Database Run IDs and All Associated Data from Database	Provides the ability for the user to delete from the database all records (i.e., RunSpecs, SprayConditions, RunConstants, IceShapeData, IceThicknessData, and ThickUtilityData records) associated with a selected set of test run IDs. To protect against inadvertent deletions, a confirmation dialog is displayed to the user prior to performing the deletion, and the user must confirm the request in order for the deletion to proceed.	Database IDs	Write Access
	Initialize/Update Conditions Data for Selected Excel File Run/Case IDs	Implements the capability for the user to add new or update existing conditions data and associated run constants for a selected set of test run/case IDs (i.e., adding or updating test run/case data in the RunSpecs, SprayConditions, and RunConstants database tables).	Excel File IDs	Write Access
	Initialize/Replace All Database Conditions Data with Values from Excel Conditions File	Enables the user to upload to the database for the first time, or replace all existing database conditions data with, conditions data from the specified Excel conditions file, calculating new or updated run constant values using the new data (i.e., updating data in the RunSpecs, SprayConditions, and RunConstants database tables). If the database already contains conditions data and/or run constants data, a confirmation dialog is displayed to the user prior to performing the update, and the user must confirm the request in order for the update to proceed.	N/A	Write Access
Conditions Data Downloading and/or Viewing	Add Conditions Data for Selected Database Run IDs to Current Excel Conditions File	Provides the capability to download from the database conditions data and run constants associated with a selected set of test run IDs (data from the RunSpecs, SprayConditions, and RunConstants tables), adding the data to the currently specified Excel conditions file.	Database IDs	Read Access
	Output Conditions Data for Selected Database Run IDs to New Excel Conditions File	Provides the capability to download from the database conditions data and run constants associated with a selected set of test run IDs (data from the RunSpecs, SprayConditions, and RunConstants tables), outputting the data to a separate, newly-created Excel conditions file.	Database IDs	Read Access
	View Excel Conditions Data File	Provides the ability for the user to select and open a conditions file in Excel.	N/A	N/A
System Maintenance Operations	Compact Database	Implements the capability to compact the database, in order to improve the database file's use of disk space and enhance the associated system performance.	N/A	Write Access
	Create & Initialize New Database File	Enables the user to create and initialize a new database file with the same specifications as the currently selected database file, including/excluding airfoil coordinate data from the original file based on the state of the "Preserve Current Airfoil Coordinate Data in New File" CheckBox.	N/A	Write Access
	View Error Log	Provides the capability for the user to view any error messages logged during the most recent attempt to upload data to the database.	N/A	N/A

without having to close and re-open the form. The "Refresh" command, in this case, is associated with the individual ListBox, and is invoked by right-clicking on the ListBox and selecting "Refresh" from the Popup menu.

Also available by right-clicking on the Database Run IDs ListBox is a command to display to the user the current size of the database file, both in megabytes, and as a percentage of the maximum file size allowed by Microsoft Access. While this information is provided automatically following any upload to the database which results in the file size being greater than 90% of the maximum size allowed by Access, by right-clicking on the ListBox and selecting "Display Database File Properties," the user is able to obtain this information at any time.

Although most of the features implemented on this form are not expected to be utilized on a routine basis, features such as the ability to delete the data associated with a specific set of run IDs, or to update conditions and constants data for a selected set of test runs, increase the flexibility of the application by enabling the user to modify database data with as fine a granularity as required. If data previously loaded into the database is later found to be invalid, or a user wishes to customize a database file by selectively adding or deleting records, this form provides the mechanism to perform such updates. When modifying the data, a user may choose to update a single value from a single test case or several different values from multiple test cases, all via one Excel spreadsheet. When performing the update, the user may choose to replace all database values with the values specified on the spreadsheet, or to only update those values which had not previously been set (i.e., null values). The choice as to which type of update to perform is made via a dialog box, displayed to the user once command execution has begun. The user is also able to choose whether to apply the selected option to the current record only, or to make the choice once, at the start of execution, and have that choice apply to all selected records.

While the database update features described above provide added flexibility at the "database record" level, the "Create & Initialize New Database File" command provides additional flexibility at the "database file" level, enabling the user to organize the data as needed, and extending the application's data management capabilities beyond the built-in limitations imposed by Microsoft Access. Utilizing this command, a user with write access is able to create a new database file, properly initialized,[§] but empty of all user-supplied data. (The user has the option to check the "Preserve Current Airfoil Coordinate Data in New File" CheckBox, in which case existing airfoil coordinate data will be retained in the new file.) As an immediate consequence of providing this capability, the amount of data that can be stored and accessed by the IceVal application is no longer constrained by the limitations of the underlying database application. If a database file should begin to approach the 2 gigabyte Microsoft Access file/table size limitation, or should become so large that system performance begins to suffer, the user can simply create a new database file and upload all future data to the newly created file. In addition, if a user wishes to create a separate database for a specific data set or group of tests (e.g., for data which is proprietary, or which requires a higher level of security), this feature likewise provides the flexibility to do so, allowing the user to easily segregate his or her data as needed.

In addition to the features included in the "database data updates" and "system maintenance operations" categories, which provide enhanced system flexibility and extensibility, the features included in the "conditions data downloading and/or viewing" category enable the creation and display of a test run data "master spreadsheet," which can be used both to review test run data contained in the database and, indirectly, to perform database updates as well.

The two frmLoadConditions CommandButtons used to download data from the database provide the capability to output to Excel all conditions and run constants data (i.e., all values included in the RunSpecs, SprayConditions, and RunConstants database tables) which are associated with a user-selected set of run IDs. Since not all RunSpecs and few RunConstants parameter values are displayed or output via other forms in the system, these CommandButtons represent the only mechanism provided within the IceVal system by which the user may view the majority of this data. As such, the primary rationale for providing this capability is to supply the user with a straightforward method for viewing and analyzing the data. However, because the data output via either of these commands is output to a spreadsheet with the same format as used by this form to read conditions data into the database, this feature can also be utilized as a first step in selectively updating conditions data. By first clicking one of these CommandButtons to output the current database values to Excel, and then editing the file in Excel, as required, the user can use the edited output file along with the database data update features to selectively update database conditions and run constants data.

frmLogin

The Database Login dialog, shown in Fig. 14, and implemented via the frmLogin.frm module, serves as the system interface to the Microsoft Access database. The logic contained in this module is used to process a user's database login, assign appropriate access privileges, and/or update the values for various database-related options. The dialog may be activated either directly, by clicking on the frmMain CommandButton labeled, "Log in to Database" (see Fig. 8, above), or indirectly, by performing an action which requires access to the database. Although logging in to the database is not required to display most forms in the system, in order to utilize the IceVal

[§]The IceVal database file "initialization" process consists of creating empty database tables for all tables included in the database, and copying the reference data required by the software, stored in the GridlineCoordinates and RefConstants database tables.

application to view and/or update data contained in the database, a valid user name and password must be entered when this dialog is displayed. If a user's attempts to log in via this dialog are unsuccessful, no database data, nor any functionality which relies upon that data, will be available.

Once a user has successfully logged in to the database via this dialog, the "View Options" CommandButton provided on the form will also be enabled. The user may then click on this CommandButton to display the extended form of the dialog.

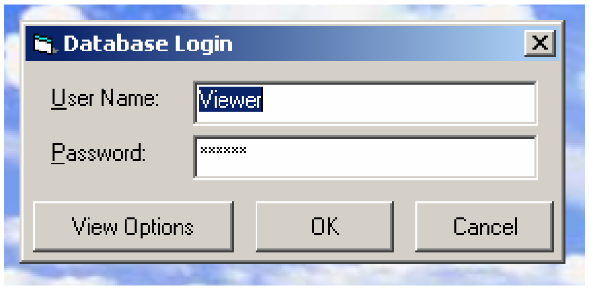


Figure 14.—Short form of the frmLogin dialog, used to log in to the database and assign access privileges.

The extended form of the frmLogin dialog, shown below in Fig. 15, is only available by accessing the dialog directly, i.e., from the main user screen (frmMain), and provides the user with the ability to view and/or update a variety of options that control a user's access to the database. Each option is managed from its own tab of a multi-tab control, and has its own set of associated CommandButtons. For this reason, so as to avoid any ambiguity which might otherwise occur, the CommandButtons at the top of the frmLogin dialog are disabled while the options display is active.

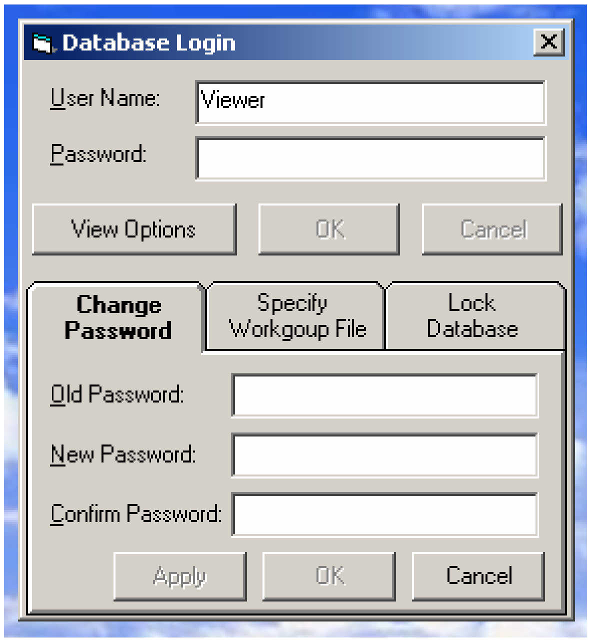


Figure 15.—Extended form of the frmLogin dialog, used to view and/or update options associated with the selected database.

As can be seen in Fig. 15, the tabbed options control consists of three tabs: "Change Password," "Specify Workgroup File," and "Lock Database." The first, and default, tab is the "Change Password" tab, which enables the user to update the account password for the currently active account (i.e., the account to which the user is currently logged in). As with most password-protected systems, in order to update the password using this form, the current account password must also be provided. This is done to protect against the possibility of a user accessing the dialog and updating the password for an account to which they were not intended to have access.

The second tab of the tabbed options control is the "Specify Workgroup File" tab. A workgroup file is associated with a specific database file, and is used by Microsoft Access to determine user accounts, passwords, access privileges, etc. For this reason, this second tab of the options display is of use primarily to database system administrators.

The third, and final, tab of the frmLogin tabbed options control is the "Lock Database" tab. This tab is provided so that a user who has logged in to the database with write access privileges may temporarily write-protect the database without having to exit the system completely or log in as a different, "read-only" user. To invoke this option, the user merely accesses the "Lock Database" tab of the frmLogin form, and clicks on the "Lock Database" CommandButton. A message is then displayed confirming to the user that the database is locked; and the database will subsequently remain write-protected until the user once again logs in with write access.

frmNewDir

The "Create Subdirectory" dialog, displayed below in Fig. 16, is accessible from either the frmAirfoilDataIO or frmDBToXL form, and may be used to create a new data output subdirectory from either form. The dialog is activated by right-clicking on the applicable form's "Output Files Directory" DirListBox, the control which both displays the directory structure for, and provides the means to update, the selected form's current output data directory. The DirListBox control is also linked to a TextBox on the same form, which displays the name of the currently selected directory. Any new directory created via this dialog will be created as a subdirectory to the directory identified in that TextBox.

Prior to creating the new subdirectory, the name supplied by the user must first be checked for validity, e.g., to ensure that no invalid characters are embedded in the name, and that the name doesn't conflict with a previously existing subdirectory. This check is performed as soon as the user clicks on the "OK" CommandButton, after entering the proposed subdirectory name in the dialog's TextBox. If the specified name is unacceptable for any reason, an error message is displayed, and the user then has the option to either provide a different name, or cancel out of the dialog by clicking on the "Cancel" CommandButton. If the name provided is valid, a confirmation dialog will appear, and will display both the name of the subdirectory to be created and the name of the directory where this new subdirectory is to be created. The user must then confirm the requested operation in order for the new subdirectory to be created.

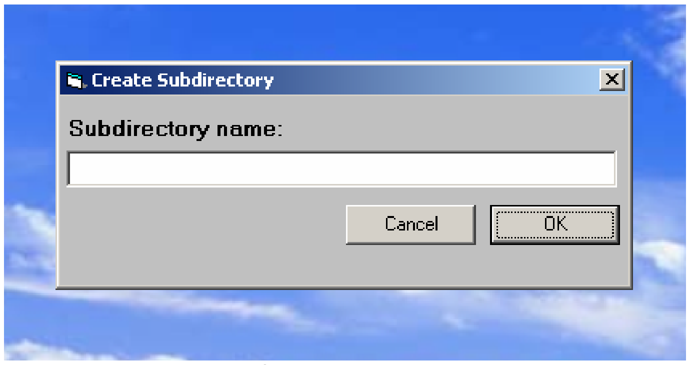


Figure 16.—The frmNewDir dialog, which enables the user to create a new subdirectory.

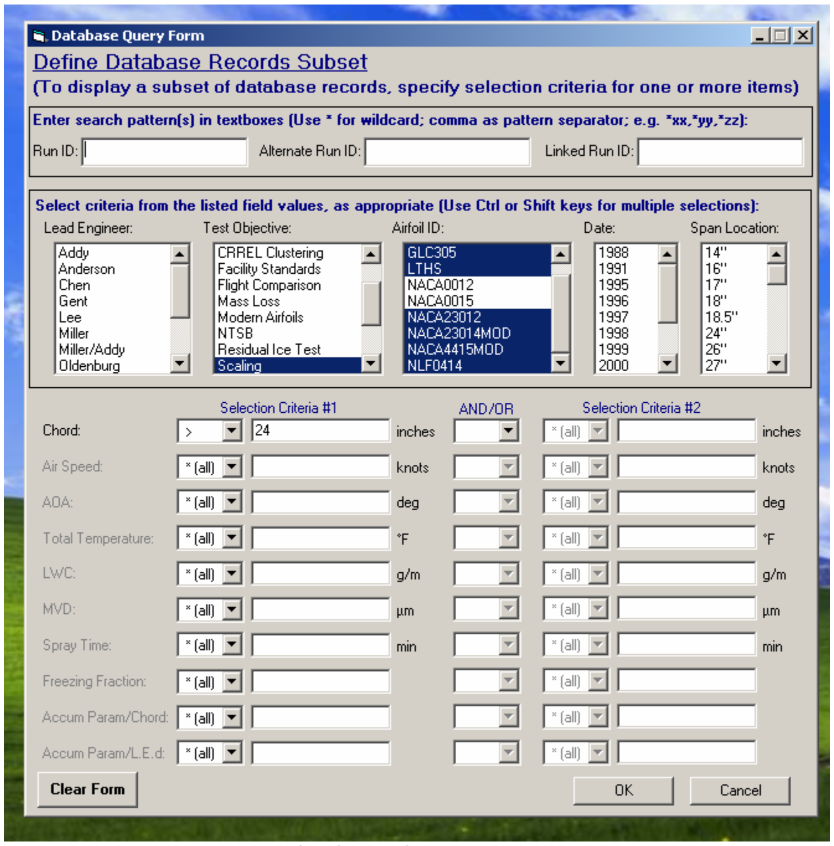
Whether a new subdirectory is successfully created or not, however, it should be noted that the parent form's currently selected output files directory will remain unchanged after exiting from this dialog. If the operation was successful, the name of the newly created directory will appear in the parent form's DirListBox. If the user wishes to utilize the new directory for data output, the directory must then be explicitly selected by double-clicking on its name in the DirListBox (See Figs. 11 and/or 12, above).

frmQuery

The final form included in the IceVal system is the modal frmQuery form, shown in Fig. 17. It is accessed from the frmDBToXL form by clicking on the "Specify Database Subset for Display" CommandButton (See Fig. 11), and its function is only fully realized in cooperation with that form.

The purpose of the frmQuery form is to provide the user with an easy-to-use, graphical means of specifying database "queries," or requests to the database to retrieve a specific data subset. To construct the query, the user must specify selection criteria utilizing one or more of the three input mechanisms supported by the form: search pattern-based Run ID TextBoxes, discrete list-based ListBoxes, and continuous range-based logical expressions, implemented via ComboBoxes and TextBoxes. After creating the query utilizing any combination of the above input methods, the user may then execute the query by clicking the "OK" CommandButton to close the form, at which point the specified database subset will then be displayed by the frmDBToXL form.

The three mechanisms utilized for specifying selection criteria on the frmQuery form have also been utilized as the foundation for the form's layout. At the top, are the


Database Query Form

Define Database Records Subset

(To display a subset of database records, specify selection criteria for one or more items)

Enter search pattern(s) in textboxes (Use * for wildcard; comma as pattern separator; e.g. *xx,*yy,*zz):

Run ID: Alternate Run ID: Linked Run ID:

Select criteria from the listed field values, as appropriate (Use Ctrl or Shift keys for multiple selections):

Lead Engineer:	Test Objective:	Airfoil ID:	Date:	Span Location:
Addy	CRREL Clustering	GLC305	1988	14"
Anderson	Facility Standards	LTHS	1991	16"
Chen	Flight Comparison	NACA0012	1995	17"
Gent	Mass Loss	NACA0015	1996	18"
Lee	Modern Airfoils	NACA23012	1997	18.5"
Miller	NTSB	NACA23014MOD	1998	24"
Miller/Addy	Residual Ice Test	NACA4415MOD	1999	26"
Oldenburg	Scaling	NLF0414	2000	27"

	Selection Criteria #1			AND/OR	Selection Criteria #2		
Chord:	>	24	inches		* (all)		inches
Air Speed:	* (all)		knots		* (all)		knots
AOA:	* (all)		deg		* (all)		deg
Total Temperature:	* (all)		°F		* (all)		°F
LWC:	* (all)		g/m		* (all)		g/m
MVD:	* (all)		µm		* (all)		µm
Spray Time:	* (all)		min		* (all)		min
Freezing Fraction:	* (all)				* (all)		
Accum Param/Chord:	* (all)				* (all)		
Accum Param/L.E.d:	* (all)				* (all)		

Clear Form OK Cancel

Figure 17.—The frmQuery form, used to set up database queries which permit the display of user-selected data subsets.

search pattern-based Run ID TextBoxes, where criteria are specified by entering a comma-separated list of search patterns into each relevant TextBox. Wildcard selections may be specified using the "*" character. In the middle of the form, are the discrete list-based ListBoxes, each of which contains a list of all values found in the current database file for the parameter specified directly above the ListBox. To specify selection criteria in this section, the user simply highlights the relevant values in each ListBox, as needed. Immediately below the ListBox section, the bottom portion of the form is devoted to the specification of continuous range-based logical expressions. Specifying selection criteria in this section is accomplished by selecting at least one relational operator from the initial ComboBox on the far left, and then entering a numeric value in the TextBox immediately to its right. If desired, a logical operator ("AND" or "OR") may also be selected from the middle ComboBox, and an additional relational operator and numeric value specified in the column labeled "Selection Criteria #2." Any invalid selection(s) made in this section of the form, such as an incomplete logical expression or a non-numeric value entered in a TextBox, will generate an error message when the user clicks the form's "OK" CommandButton. The user must then either clear the error condition or select "Cancel" in order to close the form. Finally, in the lower left corner of the form, is the "Clear Form" CommandButton. This CommandButton may be used to clear all prior selections made on the form with one button click, e.g., when the user is about to specify a new query, and the previous query consisted of a completely different set of selection criteria.

VII. Known Issues/Future Work

Although the current version of the IceVal DatAssistant software system, as described above, has met all of the original system requirements, there are, nevertheless, additional features or feature enhancements which are under consideration as potential areas for future work. These include:

1) *Implementation of an integrated on-line help system, including an electronically-accessible User's Guide*

Because the IceVal DatAssistant application was originally designed to provide a relatively limited feature set which would operate in an intuitive manner, implementation of an on-line help system was not considered a priority, and was not included among the original system requirements. As a result, the help system features currently implemented within the software are quite limited. "Tool Tips," or small popup windows that display a brief statement related to a particular control whenever the user's mouse hovers over the control, are available in selected cases throughout the system, and may be turned on or off on a form-by-form basis. In addition, informational messages are displayed on the screen in selected situations where it is particularly critical that information be provided to the user, such as when the user is about to perform an action which could cause a loss of data, or which could have other potentially harmful side-effects. While these features do provide at least a minimal level of assistance to the user, as the feature set has grown, implementation of an integrated on-line help system, including an electronically-accessible system User's Guide, has become more critical, and is now one of the top priorities for future system upgrades.

2) *Enhanced database security, with the ability to create new user accounts and/or update account access privileges directly from the IceVal application*

Access to the IceVal database is currently controlled via the specification of several pre-defined user accounts and their associated access privileges, enumerated within a special-purpose file known as a "Workgroup Information File." The access privileges assigned to these pre-existing accounts range from "read-only" access to the full privileges provided to a system administrator; and a user must log in to one of these pre-defined accounts, thereby receiving the associated level of access, in order to view or update database data. Although a user who has access to an administrative account may currently utilize the features provided within Microsoft Access to add to or modify these pre-existing accounts, or to create and assign to the database an entirely new Workgroup Information File, it is not currently possible to make these changes from within the IceVal application. Modifying the software to provide the capability for a system administrator to update user accounts and privileges via the IceVal interface, however, would offer added flexibility, and permit all system administration tasks to be accomplished directly from the IceVal application, and is therefore a potential candidate for inclusion in future software upgrades.

3) *Enhanced error handling*

While extensive error handling has been integrated into the IceVal application, as with any complex software system, it is always possible that subtle unhandled errors exist, or that unanticipated interactions may occur in the software. Over time, with increased use of the application, it is likely that any such "defects" will be detected, and that there will then be a need to enhance the system's error handling features. In addition, as users become familiar with the software, there may also be areas where user feedback dictates that the existing error handling approach be upgraded, for example, to make it more "user-friendly" or to provide additional information to the user. Although every effort had been made during the initial development process to preclude the need for any of these updates, the inherent complexities resulting from the asynchronous nature of an interactive software system greatly complicate that endeavor. In addition, error handling logic is, by definition, closely coupled to the mainline system logic, and is therefore generally impacted whenever other system modifications are made. Thus, error handling enhancements are likely to be included in future system upgrades.

4) Migration to a higher capacity, more robust and/or public domain database system, due to the limitations imposed by the use of Microsoft Access

The current implementation of the IceVal software utilizes Microsoft Access for its required database component. Although use of this database system has distinct advantages, there are also corresponding limitations, namely that the database file size is limited to 2 gigabytes, that there are potential performance/scalability issues associated with its use, and there are additional issues due to the proprietary nature of the software. As a result of these limitations, modifying the software to instead use a higher capacity and/or public domain database system is a possible update that may be incorporated into a future version of the software.

5) Implementation of user-requested enhancements

With any newly released software system, once the user community has had the opportunity to use the application and become familiar with its features, it is typical for individual users to suggest new features which might prove useful if implemented in the software. These user-requested enhancements often represent the most logical and beneficial extensions to a software system's current functionality; and, in the case of the IceVal application, these proposed upgrades will not only be given serious consideration, but will be actively sought.

6) Integration of the IceVal system into the GlennICE framework

As described elsewhere,[16] the GRC Icing Branch has recently begun an effort to develop a computational framework for all GRC icing tools. This system, referred to as the GlennICE framework, will include a common user interface which will serve both to integrate the various icing analysis modules, and to provide a common mechanism for selection of these modules as needed to perform a given task. As development of the GlennICE framework proceeds, it is anticipated that the IceVal software, along with other GRC icing tools, will be integrated into this GlennICE framework.

Conclusions

With the release of the IceVal DatAssistant icing data management system, the icing community will have access to a comprehensive, electronically-searchable ice shape database via an easy-to-use, full-function, interactive GUI. This system currently provides the ease and reliability of access, as well as the consistency of data format, to enable more thorough, less costly icing software validation, while simultaneously enhancing the potential for future upgrades to existing techniques and/or the identification of new quantitative approaches. IceVal provides experimentalists with a tool that can be used to improve experimental testing, and aircraft design engineers with a mechanism to increase the efficiency of the aircraft design process. It lays the foundation for future enhancements to GRC icing data management procedures and products, and provides the entire icing community with a tool that can facilitate the process of performing the type of broad-based, all-encompassing investigation that might one day lead to new and/or improved analytical methods.

References

[1]Olsen, W., Shaw, R., and Newton, J., "Ice Shapes and the Resulting Drag Increase for a NACA 0012 Airfoil," NASA TM–83556, Jan. 1984.

[2]Addy, H.E., Jr., "Ice Accretions and Icing Effects for Modern Airfoils," NASA/TP—2000-210031, DOT/FAA/AR–99/89, April 2000.

[3]Anderson, D.N. and Reich, A.D., "Tests of the Performance of Coatings for Low Ice Adhesion," NASA TM–107399, AIAA–97–0303, Jan. 1997.

[4]Shin, J., Bond, T.H., and Mesander, G.A., "Results of a Low Power Ice Protection System Test and a New Method of Imaging Data Analysis," NASA TM–105745, June 1992.

[5]Ide, R.F. and Oldenburg, J.R., "Icing Cloud Calibration of the NASA Glenn Icing Research Tunnel," NASA/TM—2001-210689, ARL–TR–2383, AIAA–2001–0234, March 2001.

[6]Wright, W.B., "Validation Results for LEWICE 3.0," NASA/CR—2005-213561, AIAA–2005–1243, March 2005.

[7]Vargas, M. and Reshotko, E., “Physical Mechanisms of Glaze Ice Scallop Formations on Swept Wings,” NASA/TM—1998-206616, AIAA–98-0491, Jan. 1998.

[8]Miller, D.R., Potapczuk, M.G., and Langhals, T.J., “Additional Investigations of Ice Shape Sensitivity to Parameter Variations,” NASA/TM—2006-214227, AIAA–2006–0469, March 2006.

[9]Wright, W.B., “User Manual for the NASA Glenn Ice Accretion Code LEWICE Version 2.0,” NASA/CR—1999-209409, Sept. 1999.

[10]Wright, W.B. and Rutkowski, A., “Validation Results for LEWICE 2.0,” NASA CR-1999-208690, Jan. 1999.

[11]Wright, W.B., “Users Manual for the Improved NASA Lewis Ice Accretion Code LEWICE 1.6,” NASA CR–198355, June 1995.

[12]“IEEE Standard Glossary of Software Engineering Terminology,” IEEE Std 610.12–1990, IEEE Software Engineering Standards Collection, Sept. 1994.

[13]Highsmith, J. and Cockburn, A., “Agile Software Development: The Business of Innovation,” IEEE Computer, vol. 34, no. 9, Sept. 2001, pp. 120–122

[14]McConnell, Steve, *Rapid Development: Taming Wild Software Schedules*, Microsoft Press, Redmond, Washington, 1996, Chap. 21.

[15]Anderson, David N., “Manual of Scaling Methods,” NASA/CR—2004-212875, March 2004.

[16]Wright, W.B., Potapczuk, M.G., and Levinson, L.H., “Comparison of LEWICE and GlennICE in the SLD Regime,” AIAA–2008–439, Jan. 2008.

REPORT DOCUMENTATION PAGE						*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 01-05-2008	**2. REPORT TYPE** Technical Memorandum	**3. DATES COVERED** *(From - To)*
4. TITLE AND SUBTITLE IceVal DatAssistant--An Interactive, Automated Icing Data Management System		**5a. CONTRACT NUMBER**
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Levinson, Laurie, H.; Wright, William, B.		**5d. PROJECT NUMBER**
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER WBS 457280.02.07.03.02
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191		**8. PERFORMING ORGANIZATION REPORT NUMBER** E-16399
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001		**10. SPONSORING/MONITORS ACRONYM(S)** NASA
		11. SPONSORING/MONITORING REPORT NUMBER NASA/TM-2008-215158; AIAA-2008-0443

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Categories: 05, 61, and 03
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301-621-0390

13. SUPPLEMENTARY NOTES

14. ABSTRACT
As with any scientific endeavor, the foundation of icing research at the NASA Glenn Research Center (GRC) is the data acquired during experimental testing. In the case of the GRC Icing Branch, an important part of this data consists of ice tracings taken following tests carried out in the GRC Icing Research Tunnel (IRT), as well as the associated operational and environmental conditions documented during these tests. Over the years, the large number of experimental runs completed has served to emphasize the need for a consistent strategy for managing this data. To address the situation, the Icing Branch has recently elected to implement the IceVal DatAssistant automated data management system. With the release of this system, all publicly available IRT-generated experimental ice shapes with complete and verifiable conditions have now been compiled into one electronically-searchable database. Simulation software results for the equivalent conditions, generated using the latest version of the LEWICE ice shape prediction code, are likewise included and are linked to the corresponding experimental runs. In addition to this comprehensive database, the IceVal system also includes a graphically-oriented database access utility, which provides reliable and easy access to all data contained in the database. In this paper, the issues surrounding historical icing data management practices are discussed, as well as the anticipated benefits to be achieved as a result of migrating to the new system. A detailed description of the software system features and database content is also provided; and, finally, known issues and plans for future work are presented.

15. SUBJECT TERMS
Aircraft icing; Data management systems; Relational databases; Applications programs; Data management; Data processing; Computer programs; Data retrieval; Data storage

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** STI Help Desk (email:help@sti.nasa.gov)
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	30	**19b. TELEPHONE NUMBER** *(include area code)* 301-621-0390