



A Refined Method to Translate Solar Data Quality Assessment Flags to Estimated Measurement Uncertainty

Stephen Wilcox and Thomas Stoffel

Solar Resource Solutions

NREL Technical Monitor: Aron Habte

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Subcontract Report
NREL/SR-5D00-86936
May 2024



A Refined Method to Translate Solar Data Quality Assessment Flags to Estimated Measurement Uncertainty

Stephen Wilcox and Thomas Stoffel

Solar Resource Solutions

NREL Technical Monitor: Aron Habte

Suggested Citation

Wilcox, Stephen, and Thomas Stoffel. 2024. *A Refined Method to Translate Solar Data Quality Assessment Flags to Estimated Measurement Uncertainty*. Golden, CO: National Renewable Energy Laboratory. NREL/SR-5D00-86936.

<https://www.nrel.gov/docs/fy24osti/86936.pdf>.

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Subcontract Report
NREL/SR-5D00-86936
May 2024

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Solar Energy Technologies Office. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via www.OSTI.gov.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Preface

The Data Quality and Uncertainty Integration Project is a 3-year effort to address stakeholder needs for assessing solar radiation resource data quality based on existing tools for estimating radiometer measurement uncertainties and assessing post-measurement data quality. The annual research objectives for the project address a logical progression of effort needed to achieve the project goal:

- Fiscal Year 2022—review and evaluation:
 - Evaluate existing data quality assessment methods as they relate to measurement uncertainty metrics.
 - Using existing data and simulated error conditions, develop a proof of concept for translating SERI QC flags or related information into a measure of uncertainty.
- FY 2023—conceptual development:
 - Develop a method for translating data quality assessment flags from SERI QC into estimated measurement uncertainty values.
 - Develop a method that incorporates NREL’s Solar Resource Uncertainty Application¹ and the data quality assessment uncertainty to quantify the overall uncertainty of an individual time-stamped solar radiation measurement.
- FY 2024—outreach and code development:
 - NREL will solicit industry partners for approaches to testing and the application of the newly developed code/method.
 - Develop, verify, and validate a new software package consistent with the project goal.

This technical report addresses the second objective in FY 2023, developing a refined method to translate solar data quality assessment flags into estimated measurement uncertainty.

The work presented in this report was performed under Agreement Number SUB-2022-10137 between the Alliance for Sustainable Energy, LLC and Solar Resource Solutions, LLC as part of the U.S. Department of Energy prime contract No. DE-AC36-08GO28308.

¹ See https://midcdmz.nrel.gov/radiometer_uncert.xlsx.

Acknowledgments

We are grateful to the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Solar Energy Technologies Office and to the Systems Integration and Photovoltaic subprograms for supporting this project. Specifically, we acknowledge Dr. Tassos Golnas, Dr. Guohui Yuan, and Dr. Lenny Tinker for their support and encouragement.

We also appreciate the administrative and technical support provided by Dr. Manajit Sengupta and Aron Habte in the Power Systems Engineering Center at the National Renewable Energy Laboratory.

List of Acronyms

CSV	comma separated value
DHI	diffuse (sky) horizontal irradiance
DNI	direct normal (beam) irradiance
ETR	extraterrestrial radiation
ETR _n	direct normal extraterrestrial radiation
GHI	global (total hemispheric) horizontal irradiance
GUI	graphical user interface
GUM	Guide to the Expression of Uncertainty in Measurement
ID	identification
IO	input/output
NOAA	National Oceanic and Atmospheric Administration
NREL	National Renewable Energy Laboratory
QCFIT	software for determining boundaries of acceptable solar irradiance data
RACI	responsible, accountable, consulted, and informed
SERI QC	software function for post-measurement quality assessment of solar irradiance data
SERI	Solar Energy Research Institute
SOLARUN	NREL Solar Radiometer Measurement Uncertainty application
SUNI	Solar Uncertainty Integrator
SRRL	Solar Radiation Research Laboratory
SRS	Solar Resource Solutions, LLC
SURFRAD	Surface Radiation Budget
U _o	operational uncertainty
U _{oField}	uncertainty attributable to environmental and operational effects
U _{oKt} , U _{oKn} , U _{oKd}	operational uncertainty attributable to errors during field data acquisition
U _{oSYS}	system uncertainty estimated from the three irradiance measurements (set equal to U _{oKt})
U _R	radiometer measurement uncertainty
U _{RADS}	expanded measurement uncertainty of the three radiometers
U _R GHI, U _R DNI, U _R DHI	radiometer measurement uncertainty estimated by the NREL SOLARUN application
U ₉₅ GHI, U ₉₅ DNI, U ₉₅ DHI	estimated expanded uncertainty with 95% confidence of an irradiance component determined from U _{oField} and corresponding U _R

Executive Summary

Accurate solar resource information is important for making solar energy conversion systems more cost-competitive with other energy sources. Quantifying the uncertainties of solar resource data improves the bankability, system design, and compliance of solar energy conversion systems. This project seeks to develop a method for determining the uncertainty of high-resolution solar irradiance data by incorporating results from an existing data quality assessment process with estimates of measurement uncertainty for specific radiometer design performance.

This report, the fourth of six for the Data Quality and Uncertainty Integration Project, presents a refined algorithm description for software to translate solar resource data quality assessment results into estimated uncertainty values in a Solar Uncertainty Integrator (SUNI) application. This algorithm requires three-component solar irradiance measurements—global horizontal irradiance, direct normal irradiance, and diffuse horizontal irradiance—collected at 1- to 60-minute intervals, as described in the previous deliverables.²

The uncertainty integration concept involves information available from SERI QC, a robust solar data quality assessment software tool. Developed and maintained by the National Renewable Energy Laboratory (NREL), SERI QC has been in continuous use by a variety of stakeholders for more than three decades. The other key component for determining the uncertainty of three-component solar irradiance data is the introduction of operational uncertainty (U_o) based on data from a measurement station adhering to currently accepted best practices. The concept was further developed to integrate the operational uncertainty estimated from a data quality assessment of archived solar irradiance measurements with the estimated radiometer measurement uncertainty for each of the three irradiance components to produce an estimated expanded uncertainty value with 95% confidence (U_{95}) for each data record in accordance with the accepted practices described in the ISO/IEC Guide 98-3:2008I: Guide to the Expression of Uncertainty in Measurement (ISO 2008).

This report describes in detail the overall software requirements, functional components, operational paradigm, and testing of the software system design needed to produce U_{95} values by integrating the results of the data quality assessment and radiometer measurement uncertainty. The intent of the report is to provide the development team with design concepts to bring focus and definition to the final software specifications and programming for the SUNI application.

The development of this report as Deliverable 6.4 was an iterative process that included reviews and feedback from the project team on initial drafts designed to refine how the new software could best support determining solar resource data uncertainty. The results of this effort will contribute to the final software system development by NREL staff.

² Wilcox, Stephen, and Thomas Stoffel. 2022. *Evaluation and Suitability of Using SERI QC Software for Estimating Measurement Uncertainty*. Golden, CO: National Renewable Energy Laboratory. NREL/SR-5D00-82913.

<https://www.nrel.gov/docs/fy22osti/82913.pdf>

Wilcox, Stephen, and Thomas Stoffel. 2023. *Developing the Proof of Concept for the SERI QC Flag Translation*. Golden, CO: National Renewable Energy Laboratory. NREL/SR-5D00-85589.

<https://www.nrel.gov/docs/fy23osti/85589.pdf>

Table of Contents

1	Introduction	1
1.1	The Scope of This Report.....	1
1.2	The Goal of This Report.....	1
2	Background	2
2.1	The Established Reference.....	3
2.2	Summary of Calculating U_o	3
3	Uncertainty Evaluation	5
3.1	Deriving the Uncertainties.....	5
3.2	Visualizing the Uncertainties.....	7
3.2.1	NREL Evaluation.....	7
3.2.2	SURFRAD Penn State Evaluation.....	9
3.2.3	SURFRAD Penn State Evaluation.....	11
4	Software Requirements Specifications	14
4.1	Operational Overview of the Software.....	14
4.2	Functional Specifications.....	15
4.2.1	The User Interface and File Input/Output Process.....	15
4.2.2	Uncertainty Process.....	17
4.2.3	Uncertainty Reporting Codes.....	18
4.3	Assumptions for Roles and Responsibilities.....	19
4.4	Nonfunctional Requirements.....	20
4.4.1	System Description.....	20
4.4.2	Development Software.....	20
4.4.3	Modularity.....	20
4.4.4	Error Detection and Recoverability.....	20
4.4.5	Portability.....	21
4.4.6	Security.....	21
4.4.7	Performance.....	21
4.4.8	Scalability.....	21
4.4.9	Testability.....	21
4.4.10	Accessibility.....	21
4.5	Database Requirements.....	21
4.5.1	Station Information.....	22
4.5.2	Instrument Information.....	22
4.6	Error Handling.....	22
5	Software Design Specifications	24
5.1	Program Schematic.....	24
5.2	Internal Data Structures.....	24
5.3	User Interface.....	26
5.3.1	Interface and Program Initialization.....	27
5.3.2	Main User Interface.....	27
5.4	Processing the Data.....	33
5.4.1	IO Loop.....	33
5.4.2	Output.....	37
5.4.3	Summary Statistics.....	37
5.4.4	Summary Reports.....	38
5.4.5	Uncertainty Processing.....	41
5.4.6	SERI QC.....	43
5.4.7	Uncertainty Merging.....	44
5.5	Error Handling.....	45

5.5.1	GUI Errors.....	45
5.5.2	IO Errors.....	45
5.5.3	Uncertainty Processing Errors.....	46
5.5.4	Popup Messages	47
5.6	File and Formats.....	48
5.6.1	Input File Format.....	48
5.6.2	Output File Format.....	48
5.7	Report File Format	52
5.7.1	Popup Summary Statistics.....	52
5.7.2	File Report Statistics	54
5.8	Database Specifications.....	56
5.8.1	Station Database.....	56
5.8.2	Instrument Database.....	56
5.8.3	System Configuration.....	57
5.9	Directory Structure.....	60
6	Software Testing and Validation	61
6.1	Module Functional Testing	61
6.1.1	The User Interface.....	61
6.1.2	Input File Format Errors.....	61
6.2	Functional Testing.....	62
6.2.1	Boundary Value Testing.....	62
7	Conclusions and Next Steps	63
	References	64
	Appendix A. QC0 File.....	65
	Appendix B. SERI QC Flagging Convention	66

List of Figures

Figure 1. NREL data 2021	8
Figure 2. Expanded detail of the U _{RADS} trace in Figure 1	9
Figure 3. Fort Peck data 2021	10
Figure 4. Penn State data 2021	12
Figure 5. Schematic overview of system	15
Figure 6. Main GUI.....	28
Figure 7. Sample popup summary statistics report	53

List of Tables

Table 1. NREL Estimated Radiometer Measurement Uncertainty Values from the SOLARUN Application.....	7
Table 2. NREL Average Values	9
Table 3. SURFRAD Radiometer Uncertainty Estimated Values from the SOLARUN Application	9
Table 4. Fort Peck Average Values	11
Table 5. Penn State Average Values.....	13
Table 6. Additional SERI QC Return Values	17
Table 7. Uncertainty Status Codes.....	19
Table 8. Summary of Key Activities, Roles, and Responsibilities	20
Table 9. Error Conditions and Resolutions	23
Table 10. SERI QC Return Codes	43
Table 11. Defined IO Errors	46
Table 12. Defined Errors from Uncertainty Process.....	46
Table 13. Input File Format	48
Table 14. Standard Output File Header	49
Table 15. Standard Output File Format	50
Table 16. Extended Output File Header.....	51
Table 17. Extended Output File Format.....	52
Table 18. Popup Summary Statistics	53
Table 19. Report File Format.....	55
Table 20. Instrument Table Record Description	57
Table 21. Configuration File Format and Contents	59
Table B-1. SERI QC Flagging Convention	66

1 Introduction

1.1 The Scope of This Report

This report is the fourth of six deliverables in the Data Quality and Uncertainty Integration Project assigned to Solar Resource Solutions, LLC and presents a refined algorithm description for software to translate solar resource data quality assessment results and radiometer measurement uncertainty estimates into integrated uncertainty values. This Solar Uncertainty Integrator application (SUNI) requires three-component solar irradiance measurements collected at 1- to 60-minute intervals, as described in Deliverable 6.1 (Wilcox and Stoffel 2022) and Deliverable 6.2 (Wilcox and Stoffel 2023).

1.2 The Goal of This Report

The results of this effort will provide an overview of what need the software addresses, a description of likely users and how they will interact with the software, functional and non-function requirements, the software processing algorithm, the system output options, proposed methods of validation, and a description and roles of participants in the project.

This document will have the primary focus of providing operational concepts to National Renewable Energy Laboratory (NREL) project participants to describe functional boundaries and options for the new software system. Feedback from the reviews and discussions by the project team of the previous reports has been incorporated into this deliverable, which will be the final system and software specifications for NREL developers.

2 Background

Methods for estimating the measurement uncertainty of pyrhemometers and pyranometers have been available for several decades (WMO 1971; Iqbal 1983; ISO 2008). Similarly, solar irradiance data quality assessment tools have been developed by various researchers (Maxwell et al. 1993; Shi and Long 2003). Deliverable 6.1 examined the feasibility of using the existing NREL data quality assessment package SERI QC as the foundation for developing the concept of an operational uncertainty (U_o), uncertainty attributable to errors during field data acquisition. Necessary are three-component solar irradiance measurements using information available from SERI QC, which will be incorporated with the NREL radiometer measurement uncertainty (U_R) from the NREL Solar Radiometer Measurement Uncertainty (SOLARUN) application (Habte 2014) to determine the final uncertainty estimates for a record of solar measurements. The results of that investigation determined that SERI QC, as a well-established, robust, and widely used software, is well-positioned to provide the basis for this project. As a software function that operates on a single record of data, SERI QC, with minor modification, can be used as the core processing module to provide front-end processing, flag evaluation, and ancillary data to the U_o process.

The U_o method described and validated in Deliverable 6.2 (Wilcox and Stoffel 2023) requires the use of simultaneous measurements of global horizontal irradiance (GHI), direct normal irradiance (DNI), and diffuse horizontal irradiance (DHI). The process is sensitive enough to limit its scope to data carefully collected with protocols and guidance detailed in the best practices manual (Sengupta et al. 2021). In this context, the process is not intended to discriminate between, for example, bad data and very bad data. Rather, it is intended to discriminate between good data and data with typical and occasional operational errors. Its purpose is to provide an intelligent estimate of uncertainty for solar irradiance data with a foundation of excellence in measurement protocols, such as would be demanded by analysts evaluating the viability of a proposed solar power project.

Deliverable 6.3 reported on the iterative process to further develop the algorithm based on reviews and feedback from the project team on how the new software could best support determining solar resource data uncertainty by integrating data quality assessments and radiometer measurement uncertainties.

Central to the data quality assessment method, the measured values of GHI, DNI, and DHI are converted by SERI QC to clearness indices in K-space, which is a normalized representation of an irradiance measurement, based on the corresponding values of extraterrestrial irradiance. K-space values generally range³ from 0 to 1 and reveal atmospheric clarity independent of the station location and time of day (see Section 3.1 in Deliverable 6.2 for a detailed derivation [Wilcox and Stoffel 2023]). These measurements are related by the coupling equation in K-space:

$$K_t = K_n + K_d \tag{1}$$

³ K-space values of normalized solar irradiance data can exceed 1.0 for very brief periods due to the relative positioning of the solar disk and select clouds responsible for the cloud enhancement of a measurement.

where:

- K_t is the normalized K-space value of GHI.
- K_n is the normalized K-space value of DNI.
- K_d is the normalized K-space value of DHI.

2.1 The Established Reference

Equation 1 is the measurement equation used to determine the operational uncertainty (U_o) by establishing a field reference for each component through the other two components in the coupling equation. By rearranging Eq. 1, an operational uncertainty ratio can then be calculated, which with “perfect data” will equal one. Any measurement error would result in a ratio less than or greater than one. With Eq. 2, Eq. 3, and Eq. 4, a percentage error is calculated to determine the operational uncertainties for the three components:

$$U_o K_t = \left(\frac{K_t}{K_n + K_d} - 1 \right) \cdot 100 \quad (2)$$

$$U_o K_n = \left(\frac{K_n}{K_t - K_d} - 1 \right) \cdot 100 \quad (3)$$

$$U_o K_d = \left(\frac{K_d}{K_t - K_n} - 1 \right) \cdot 100 \quad (4)$$

The proof of concept described in Deliverable 6.2 (Wilcox and Stoffel 2023) established that Eq. 3 and Eq. 4 are undefined if $K_t = K_d$ or $K_t = K_n$. Certain other measurement conditions will result in exaggerated and unrealistic results from the same equations. Additionally, measurements that occur at high zenith angles (near sunrise and sunset) can result in similar unrealistic values from ratios between small numbers. Thus, it was determined that $U_o K_n$ and $U_o K_d$ can be applied only in specifically restricted data scenarios. As a result, this method cannot assign a U_o for every data record in a typical data set because any records with measurements producing these conditions either cannot or should not be used in an uncertainty calculation.

To help alleviate this restriction, further investigation, described in Deliverable 6.2 (Wilcox and Stoffel 2023), established that $U_o K_t$ adequately captures errors in DNI and DHI, and thus $U_o K_t$ can be used nearly universally to represent operational uncertainty in all three components. Further, the investigation established that $U_o K_n$ defined by Eq. 3 can be calculated without the vulnerabilities posed by small numbers in the denominator if restricted only to measurements collected during very clear skies, which are also the atmospheric conditions representing the greatest energy contribution to solar power generation. Likewise, under overcast skies, $U_o K_d$ defined by Eq. 4 is more appropriately calculated free from interference from small numbers in the denominator.

2.2 Summary of Calculating U_o

The operational uncertainty estimates for global irradiance, $U_o K_t$, will be used as a reasonable proxy for all three components. Although this approach will not provide specific uncertainties for each component, it is an unavoidable circumstance. In all cases described, regardless of the method of calculation, the nature of the uncertainty equations and the use of field data are such

that each is subject to crossover errors from the other two components. We call this the *ambiguity of fault*, or the general inability to determine which of the three components causes an imbalance in an uncertainty equation. Nonetheless, U_{OKt} as an overarching uncertainty can be assigned for the entire data record, which embodies a valuable contribution to estimating the uncertainty of a data set.

Given these methods and restrictions, the remainder of this report provides a description of the software and application environment needed to calculate and report U_{95} for a data record of GHI, DNI, and DHI by integrating data quality assessments (U_O) and radiometer measurement uncertainties (U_R).

3 Uncertainty Evaluation

The goal of this process is to apply the ISO/IEC Guide 98-3:2008I: Guide to the Expression of Uncertainty in Measurement (GUM) (ISO 2008) to the integration of radiometer measurement and operational uncertainties to estimate the expanded uncertainty with 95% confidence (U_{95}) for historical measurements of the three fundamental elements of solar irradiance: GHI, DNI, and DHI.

Proper accounting of the expanded uncertainty of the radiometer types as calibrated and the operational uncertainties of the radiometers deployed for field measurements is needed to assign uncertainties to the archived irradiance data. Determining the final uncertainty will require including the radiometer measurement uncertainties themselves.

3.1 Deriving the Uncertainties

The first step establishes an expanded radiometer measurement uncertainty for each irradiance component, which is accomplished with the NREL SOLARUN application (Habte 2014) (external to this process). This uncertainty represents the bounds of the measurement error based on the design performance specifications of the radiometer and operated under well-controlled conditions, such as a calibration scenario. In this evaluation, we label these as the radiometer measurement expanded uncertainties, U_R :

- Uncertainty of GHI: U_{RGHI}
- Uncertainty of DNI: U_{RDNI}
- Uncertainty of DHI: U_{RDHI} .

Beyond these radiometer uncertainties, we expect adverse environmental or operational field conditions to *add* to the uncertainty of a measurement. Through the uncertainty equations 2, 3, 4, we have an estimate of *all* uncertainties present in the field measurement system. As noted previously, we will rely on U_{OKt} to represent operational errors in all three components. *For this evaluation and here on, we will relabel U_{OKt} as a system uncertainty, U_{OSYS} , which has relevance to all three components.*

In the context of field measurements, U_{OSYS} comprises two discrete, nonoverlapping components: (1) the radiometer measurement uncertainties and (2) uncertainties attributable to field operations. To separate the field uncertainty from the radiometer expanded uncertainties, first, we establish the collective expanded uncertainty for the radiometers (U_{RADS}) by combining the three radiometer measurement uncertainties in quadrature:

$$U_{RADS} = 2 \cdot \sqrt{\left(\frac{U_{RGHI}}{2}\right)^2 + \left(\frac{U_{RDNI}}{2}\right)^2 + \left(\frac{U_{RDHI}}{2}\right)^2} \quad (5)$$

Equation 5 is consistent with the GUM principles in that the three U_R values are assumed to have normal distributions, i.e., each expanded uncertainty is divided by 2 to represent the combined uncertainties, and a coverage factor (k) of 2 is applied to the radical to yield an expanded uncertainty with a 95% confidence interval. Using the approach described in Myers et al. (2000) for apportioning uncertainty estimates according to levels of irradiance and knowing that the contribution of K_n and K_d in the denominator of Eq. 2 commonly differ by an order of

magnitude, instead, we can rewrite Eq. 5 with uncertainties *proportional to the contribution of each irradiance component* in the denominator of Eq. 2:

$$U_{RADS} = 2 \cdot \sqrt{\left(\frac{UrGHI}{2}\right)^2 + \left(\frac{UrDNI \cdot Kn_frac + UrDHI \cdot Kd_frac}{\sqrt{3}}\right)^2} \quad (6)$$

where:

$$Kn_frac = \frac{Kn}{Kn+Kd} \quad (7)$$

and:

$$Kd_frac = \frac{Kd}{Kn+Kd} \quad (8)$$

(Note: $Kn_frac + Kd_frac = 1$.)

The proportional irradiance factors in Eq. 7 and Eq. 8, which are the result of variable sky conditions, will affect the distribution of U_{RDNI} and U_{RDHI} . Thus, because there is no definite knowledge of the possible distribution of these factors under all sky conditions, in this case, it is the best and conservative estimate to assume a rectangular distribution, i.e., by dividing the uncertainties by the square root of 3 to determine the standard uncertainties of the pyrheliometer and the shaded pyranometer, as shown in Eq. 6. Note that the whole of Eq. 6 shows the expanded uncertainty of the radiometer uncertainties by combining the three-component uncertainty and applying the desired coverage factor ($k = 2$, ~95% confidence level).

U_{oField} , the uncertainty attributable to environmental and operational effects, is then the difference between the absolute value of U_{oSYS} and U_{RADS} :

$$U_{oField} = MAX[|U_{oSYS}| - U_{RADS}, 0] \quad (9)$$

(Negative values of U_{oField} are set to zero.)

In this arrangement, U_{oField} represents the uncertainty in U_{oSYS} *beyond* that of the estimated radiometer uncertainties (U_{RADS}). Combining U_{oField} (if any) with each radiometer uncertainty in turn, we arrive at estimates of the expanded uncertainty for each of the three measured irradiance components:

$$U_{95GHI} = 2 \cdot \sqrt{\left(\frac{UrGHI}{2}\right)^2 + \left(\frac{U_{oField}}{2}\right)^2} \quad (10)$$

$$U_{95DNI} = 2 \cdot \sqrt{\left(\frac{UrDNI}{2}\right)^2 + \left(\frac{U_{oField}}{2}\right)^2} \quad (11)$$

$$U_{95DHI} = 2 \cdot \sqrt{\left(\frac{UrDHI}{2}\right)^2 + \left(\frac{U_{oField}}{2}\right)^2} \quad (12)$$

These three equations (1) preserve the radiometer uncertainty, which is the lowest assignable limit of uncertainty in a measurement, and (2) include all other uncertainties in U_{oSYS} that exceed the combined radiometer uncertainties.

3.2 Visualizing the Uncertainties

The NREL/Solar Radiation Research Laboratory (SRRL) and National Oceanic and Atmospheric Administration (NOAA)/Surface Radiation Budget (SURFRAD) Fort Peck and Penn State 2021 proof-of-concept data described in Deliverable 6.2 (Wilcox and Stoffel 2023) were subjected to a simplified version of this approach to visualize the values and interactions of the interim values. Equations 2 and 6–12 were used for this evaluation.

3.2.1 NREL Evaluation

The values for U_{RGHI} , U_{RDNI} , and U_{RDHI} were estimated from the NREL SOLARUN application based on the radiometer manufacturer and model (see Table 1).

Table 1. NREL Estimated Radiometer Measurement Uncertainty Values from the SOLARUN Application

Parameter	Instrument	Value (%)
U_{RGHI}	Kipp & Zonen CMP22	± 3.5
U_{RDNI}	Kipp & Zonen CHP1	± 2.3
U_{RDHI}	Kipp & Zonen CMP22	± 3.5

Measurements at SRRL for 2021 were used to compute the values of U_{RGHI} , U_{OSYS} , U_{RADS} , and U_{95GHI} and are plotted in Figure 1. The apparent thickness of the U_{RADS} gray trace indicates its variability due to applying proportional uncertainties (Eq. 6). An expanded view of U_{RADS} is shown in Figure 2 indicating the effect of the proportional uncertainties. The averages of the values plotted in Figure 1 are shown in Table 2, along with the percentage of times the U_{95GHI} values exceeded U_{RGHI} (one measure of operational uncertainty resulting from an unknown combination of the measurement conditions that differ from those of the calibration, radiometer installation, and effectiveness of equipment maintenance).

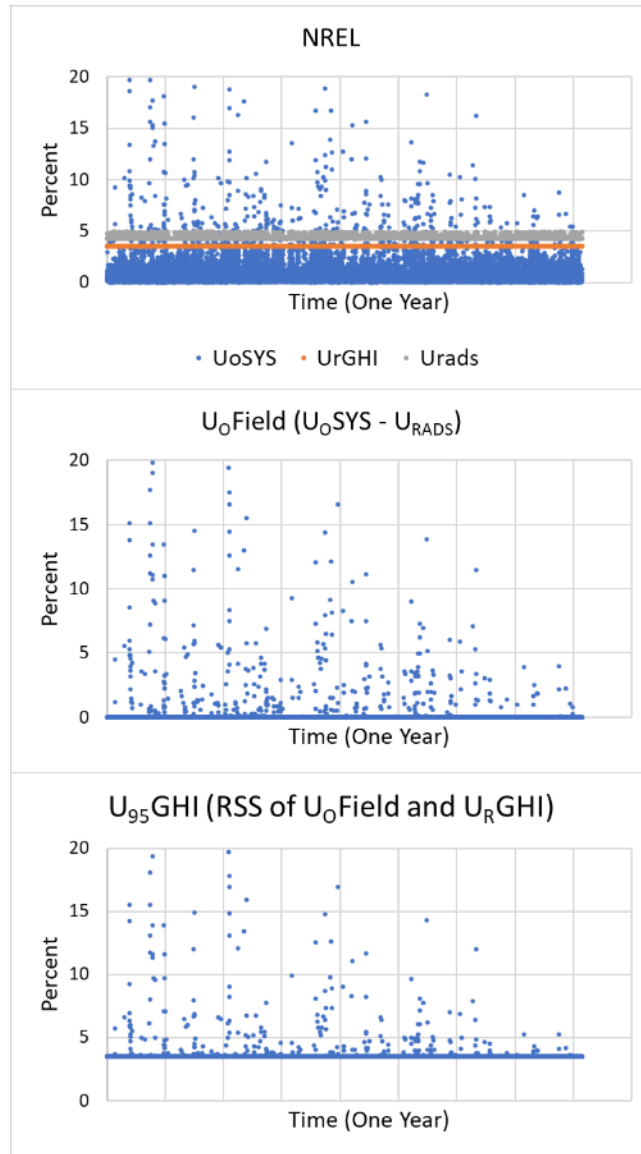


Figure 1. NREL data 2021

Top: U_oSYS (blue), U_RGHI (orange), U_{RADS} (gray); middle: U_oField; bottom: U₉₅GHI

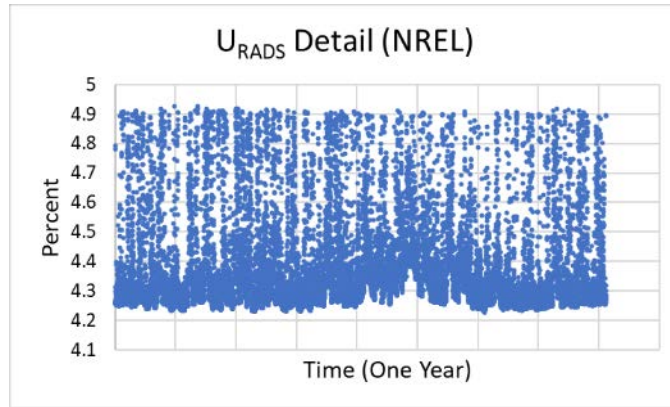


Figure 2. Expanded detail of the U_{RADS} trace in Figure 1

Variability is due to the proportioned U_{RDNI} and U_{RDHI} uncertainties (Eq. 6) caused by varying irradiance

Table 2. NREL Average Values

Parameter	Value (%)
$U_{\text{O}}\text{SYS}$	± 1.04
$U_{\text{O}}\text{Field}$	± 0.12
$U_{95}\text{GHI}$	± 3.58
U_{RADS}	± 4.39
$U_{95}\text{GHI}$ exceeds U_{RGHI}	2.3

3.2.2 SURFRAD Penn State Evaluation

Based on the makes and models of the radiometers used at this NOAA station, the values for U_{RGHI} , U_{RDNI} , and U_{RDHI} were estimated from the SOLARUN application (see Table 3).

Table 3. SURFRAD Radiometer Uncertainty Estimated Values from the SOLARUN Application

Parameter	Instrument	Value (%)
U_{RGHI}	Spectrolab SR-75 ⁴	± 4
U_{RDNI}	Eppley NIP	± 2.5
U_{RDHI}	Eppley 8-48	± 3.5

Measurements at this NOAA station for 2021 with 1-minute resolution were used to compute values for U_{RGHI} , $U_{\text{O}}\text{SYS}$, U_{RADS} , and $U_{95}\text{GHI}$ and are plotted in Figure 3. The apparent thickness of the U_{RADS} gray trace indicates its variability due to applying proportional

⁴ The manufacturer's performance specifications for the Spectrolab model SR-75 are from Coulson (1975).

uncertainties (Eq. 6). The effect is like that demonstrated using NREL data in Figure 2. The averages of the values plotted in Figure 3 are shown in Table 4, along with the percentage of times the $U_{95}GHI$ values exceeded U_{RGHI} . The latter is one measure of operational uncertainty, but note that despite the significantly higher value than the same statistic for NREL, the average $U_{95}GHI$ is only slightly higher than the U_{RGHI} ($\pm 4.24\%$ vs. $\pm 4\%$). This indicates that the station is largely operating within the uncertainties of the GHI radiometer. The comparisons for the DNI and DHI values would be similar.

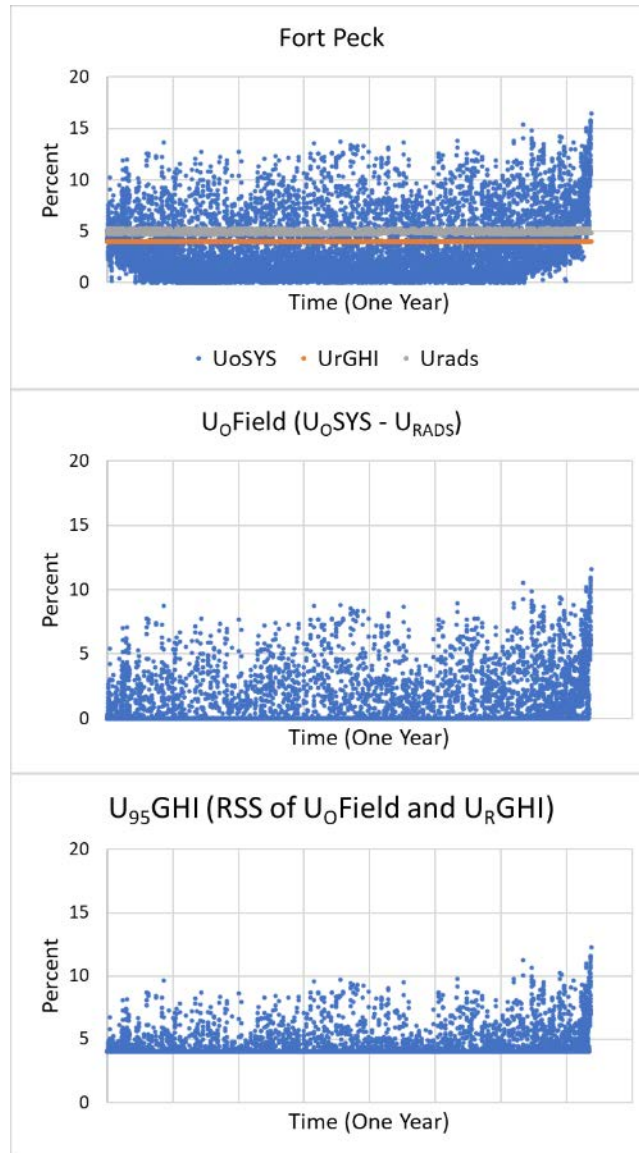


Figure 3. Fort Peck data 2021

Top: U_{oSYS} (blue), U_{rGHI} (orange), U_{rads} (gray); middle: U_{oField} ; bottom: $U_{95}GHI$

Table 4. Fort Peck Average Values

Parameter	Value (%)
U _o SYS	±2.96
U _o Field	±0.58
U ₉₅ GHI	±4.24
U _{RADS}	±4.92
U ₉₅ GHI exceeds U _R GHI	20.4

3.2.3 SURFRAD Penn State Evaluation

The values for U_RGHI, U_RDNI, and U_RDHI were set to constant values estimated from the NREL SOLARUN application for the same radiometers used at Fort Peck and as shown in Table 3.

Measurements at this NOAA station for 2021 were used to compute values for U_RGHI, U_oSYS, U_{RADS}, and U₉₅GHI, as plotted in Figure 4. The apparent thickness of the U_{RADS} gray trace indicates its variability due to applying proportional uncertainties (Eq. 6). The effect is demonstrated using NREL data in Figure 2.

The averages of the values plotted in Figure 4 are shown in Table 5, along with the percentage of times the U₉₅GHI values exceeded U_RGHI. The latter is one measure of operational uncertainty, but note that despite the significantly higher value than the same for NREL, the average U₉₅GHI is only slightly higher than the U_RGHI (±4.38% vs. ±4%). This indicates that the station is largely operating within the uncertainties of the GHI radiometer. The comparison for the DNI and DHI values would be similar.

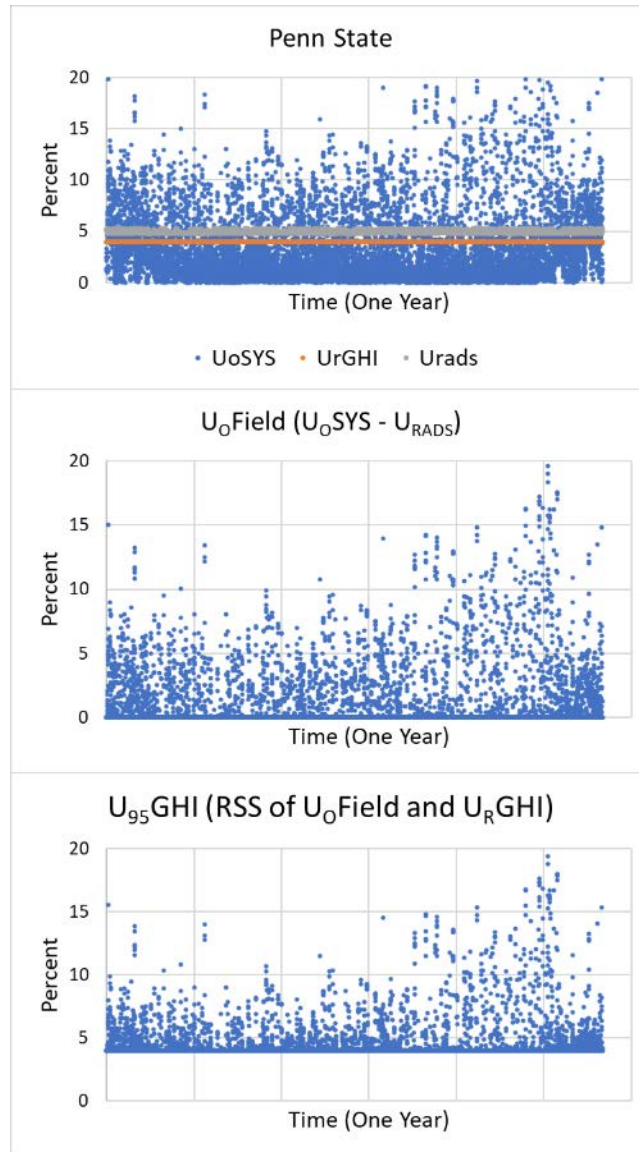


Figure 4. Penn State data 2021

Top: U_oSYS (blue), U_RGHI (orange), U_{RADS} (gray); middle: U_oField; bottom: U₉₅GHI

Table 5. Penn State Average Values

Parameter	Value (%)
U _o SYS	±3.16
U _o Field	±0.76
U ₉₅ GHI	±4.38
U _{RADS}	±4.93
U ₉₅ GHI exceeds U _R GHI	21.3

4 Software Requirements Specifications

To adequately evaluate the financial and technical viability of a proposed solar energy conversion project, the data used to characterize the solar resource must deliver a measure of resource magnitude and an estimate of its uncertainty. Without this knowledge, estimates of a facility's ability to convert a given quantity of solar energy to other forms of energy cannot be properly compared to the boundaries of financial feasibility. This SUNI software provides an automated process to estimate radiometer measurement uncertainty (U_{95}) based on the NREL SOLARUN application (Habte 2014) combined with the operational uncertainty (U_o) calculations from field data as described in the previous section.

These software requirements specifications are intended for two audiences:

1. Expert project principals who are knowledgeable about the overall goals and intended use of the software and with detailed knowledge of the uncertainty processes involved
2. The software development team that will provide the design and coding to fulfill the requirements.

The specifications outlined in Deliverable 6.3 have been reviewed by all participants, and now they are refined and finalized in this report (Deliverable 6.4) in preparation for the final coding phase.

4.1 Operational Overview of the Software

The SUNI application provides users with a method to assign expanded uncertainty estimates to measured solar radiation data. The system will merge static uncertainty information about a radiometer with the dynamic operational uncertainty information extracted from the measurements themselves, according to Section 3. The design assumes that the user has available a data file and supporting metadata from a measurement station for which it is desired to attach an uncertainty estimate to each measurement value. Although it is not a part of this system, an analyst might use the output information in a variety of ways, ranging from an immediate assessment of the quality of a particular measurement to aggregated summaries to understand the accuracy and applicability of the measured solar resource over time at that location. The effectiveness of station operations and maintenance practices can also be demonstrated by the new process results.

The software will facilitate the processing of a data file containing GHI, DNI, and DHI measurements recorded at intervals from 1 minute to 60 minutes over any reasonable period (e.g., days, months, years, or decades—limited only by the capacity and processing capabilities of the system). A graphical user interface (GUI) allows entry of the name of an input data file (which holds time and date information and the three solar irradiance measurements), measurement station information (including location), a list of predetermined uncertainty parameters for each radiometer, and the name of an output file. Integral to the process is the SERI QC data quality assessment function, which provides the necessary data file input validation, format verification, and quality flags to determine which, if any, of the uncertainty algorithms will be invoked.

On demand by the user, the software will read the input file, examine the three GHI, DNI, and DHI components on a record-by-record basis, and return SERI QC flags and uncertainty values.

These values will be written to the output file to be available for further analyses external to this system. Figure 5 provides a schematic overview of the system design.

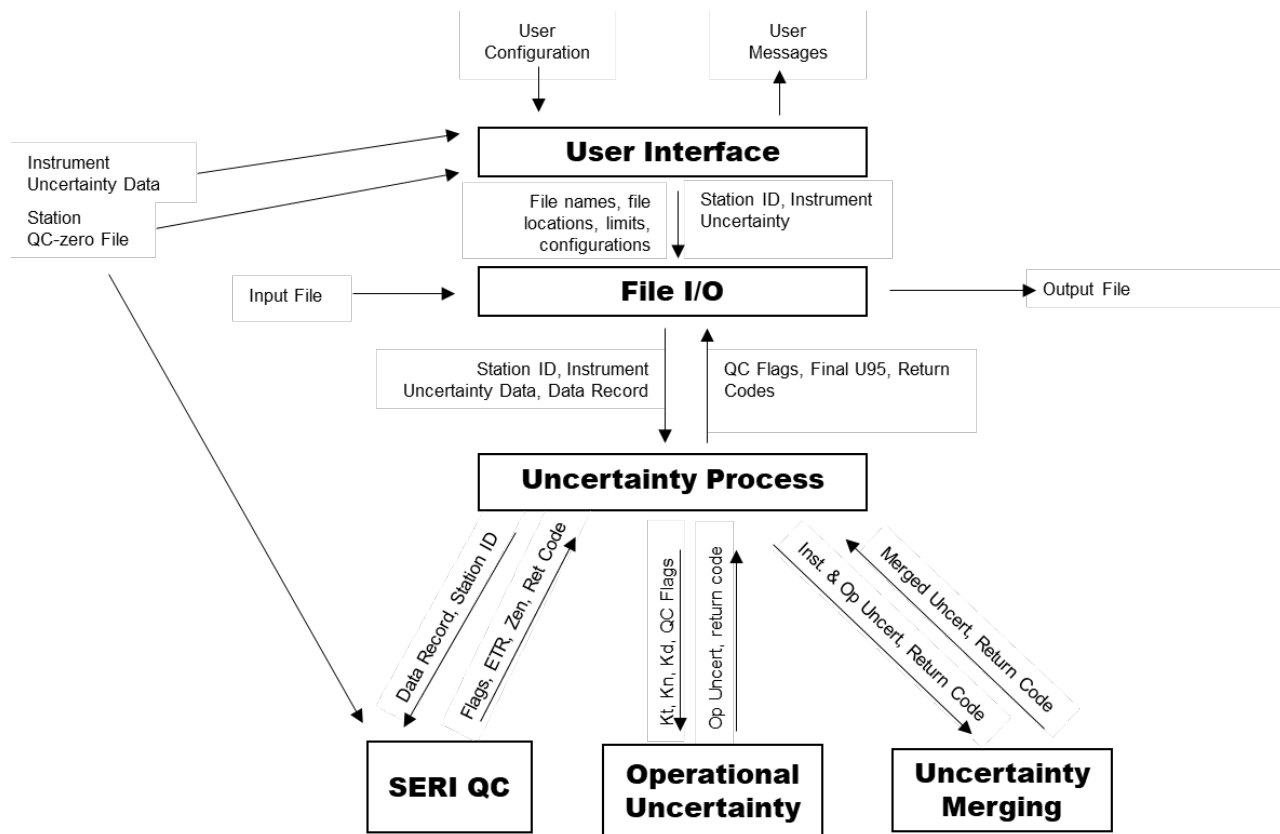


Figure 5. Schematic overview of system

4.2 Functional Specifications

Although the overarching software design described here will process an input data file of solar irradiance measurements and create an output file containing the resulting SERI QC flags and combined uncertainty values, the design will be modular to allow for the incorporation of the core uncertainty routines in other processing environments. This design facilitates future uses of the uncertainty algorithm for other targeted software solutions (e.g., within an automated acquisition or quality assessment data stream).

4.2.1 The User Interface and File Input/Output Process

The top-level user interface—the umbrella process—will allow an analyst to enter information required for the system to locate the input and output data files, retrieve the station’s geographic attributes, and retrieve uncertainty information for each radiometer (i.e., the parameters needed to compute U_R for each type of radiometer based on the NREL SOLARUN application).

The required information is described as follows:

- **Input data file**—the name and path of the file holding the solar irradiance measurements in the input file format. The file format will be non-configurable as a comma separated value (CSV) format compatible with the Microsoft Excel CSV (comma delimited) (*.csv)

export format. The date format will be magnitude sort order (YYYY-MM-DD), an option in the Excel export format. Time will be in 24-hour format (HH:MM).

- **SERI QC QC0 file**—a file in the .qc0 format (Maxwell et al. 1993) (see Appendix A) that provides:
 - Station name
 - Station latitude and longitude
 - Station time zone
 - Expected value data boundaries determined empirically for the station using results from QCFIT (a boundary-fitting companion application with SERI QC).
- **Radiometer uncertainty parameters**—From a database holding results from the NREL SOLARUN application, the user will enter the identifiers for the three instruments in use, which will retrieve the expanded measurement uncertainty estimates (percentage of reading) for each of the three instruments.
- **Configuration specifications**—specific parameters that limit or modify the bounding conditions for calculating the operational uncertainty. Several parameters might have some degree of sensitivity to local climatological influences or even the intended use of the uncertainty results (e.g., limiting or expanding the scope of the data under consideration). These parameters will allow the user to modify the system defaults for selecting data records to be processed:
 - Minimum DNI threshold
 - Minimum GHI threshold
 - Maximum zenith angle threshold
 - Maximum allowable SERI QC flag.
- **Output data file**—the name and path of the file that is to receive the SERI QC flags, uncertainty values, and status code in the output file format. The output file format will be non-configurable as a CSV format compatible with the Microsoft Excel CSV (comma delimited) (*.csv) export format.

The user interface will assemble all necessary input information and then pass it to the uncertainty module to invoke processing for each record (see Figure 5).

Once the configuration is complete, the process will begin the file input/output (IO) loop, which will read a data record and pass it and other configuration information to the uncertainty module for evaluation. The IO loop will provide a progress meter to inform the user of processing progress.

The uncertainty module will work independently to process a single data record. Upon error or exception, it will cleanly halt the process and return the appropriate status message to the main IO process.

Upon completion of the uncertainty process, the main IO process will interpret and display (if appropriate) return values for the user. Depending on the returned value, the IO process will continue with the next record, or if an error condition is reported, it will provide continuation options to the user.

Upon completion of reading the input file, the main IO process will display a summary of processing and uncertainty statistics for consideration by the user.

4.2.2 Uncertainty Process

All necessary processing information and input data will be passed to the uncertainty process module from the IO process. The uncertainty module is designed as a complete function to process a record of measurement data and return the data uncertainties and process status codes. The uncertainty module consists of three submodules:

- SERI QC
- Operational uncertainty
- Uncertainty merging.

The uncertainty process will receive a data record (GHI, DNI, DHI) from the umbrella process along with a pointer to the SERI QC QC0 file (see Section 4.2.2.1) and radiometer uncertainty parameters. Each uncertainty submodule will be called in turn from the uncertainty process.

4.2.2.1 SERI QC

SERI QC is well-documented in Maxwell et al. (1993). It is a software function designed to process a single record of solar irradiance data when called.

For the SUNI application, SERI QC will require modification to return additional information to the uncertainty calling routine via the modified SERI QC function prototype. These variables (Table 6) are created internally during the execution of SERI QC and will be returned by reference in three additional function parameters.

Table 6. Additional SERI QC Return Values

Parameter	Data Type
Solar zenith angle (refracted)	float
ETR	float
ETRn	float

When invoked, the existing SERI QC function initially performs a sequence of data validation steps to ensure that the input data, date, and time are within necessary bounds. With any detected error condition, SERI QC returns an error code to the calling module and processing stops. Any code(s) will be returned to the umbrella module for interpretation, generation of a message to the user, and a presentation of user options for continuing.

If SERI QC completes the evaluation of the record and successfully returns flags, the GHI flag will be evaluated first to determine if the input record consisted of valid three-component data. Data are valid for passing to the uncertainty calculations if the following conditions are met, which represent a valid three-component SERI QC flag:

1. Flag = 3 OR
2. Flag = 9 OR

3. Flag ≥ 10 and ≤ 89 and $((\text{Flag}+2) \bmod 4) < 2$.

(See Appendix B for a full description of SERI QC flags.)

In any other flag scenario, the following uncertainty calculations will not be performed, and an appropriate code will be returned to the calling IO routine (see Table 7).

SERI QC requires a QC-zero file created by the separate program QCFIT (Maxwell et al. 1993) (Appendix A). This file must already exist in a user-specified directory prior to running this process.

4.2.2.2 Operational Uncertainty (U_O)

The operational uncertainty routine will be called with a parameter list consisting of GHI, DNI, and DHI, and from SERI QC, the values of ETR and ETRn. The K-space variables of K_t , K_n , and K_d will be calculated using the corresponding extraterrestrial values as described in Section 3.1 of Deliverable 6.2 (Wilcox and Stoffel 2023):

The calculation of the U_O using the uncertainty Eq. 2 will proceed in this order according to the equations in Section 3.1:

- If $\text{DNI} > 25 \text{ W/m}^2$ (or other configurable minimum DNI), $U_{O\text{SYS}}$ ($U_{O\text{Kt}}$) will be calculated using Eq. 2, and a $U_{O\text{SYS}}$ success status flag (Table 7) will accompany the value. If $\text{DNI} \leq 25 \text{ W/m}^2$ (or other configurable minimum), no further uncertainty calculations will be made, and a missing value and flag for the uncertainty field will be assigned (Table 7).
- The radiometer uncertainty (U_R) for each of the three components will be retrieved from the instrument measurement/calibration uncertainty database. These values will be retrieved only once during any processing of a data file because they are constant for each specific radiometer model.
- The U_{RADS} will be calculated according to Eq. 6 using the three individual U_R values retrieved from the instrument uncertainty database.
- $U_{O\text{Field}}$ will be calculated using $U_{O\text{SYS}}$ and U_{RADS} according to Eq. 9.

4.2.2.3 Final Combined Uncertainty for Each Component

The three U_{95} values for each irradiance measurement will be calculated according to Eq. 10–12. Uncertainty status codes and return values (see Section 4.2.3) will be assigned according to the status of the uncertainty calculations described in this section.

4.2.3 Uncertainty Reporting Codes

The uncertainty values (U_{95}) generated by the uncertainty merging function (Fig. 5) will be returned to the uncertainty process. In the uncertainty process, a status code will be assigned to each uncertainty field ($U_{95\text{GHI}}$, $U_{95\text{DNI}}$, $U_{95\text{DHI}}$) indicating if the value was successfully generated, and if not, the cause of the failure. Table 7 shows the codes and conditions.

Table 7. Uncertainty Status Codes

Code	Condition	U₉₅ Value
0	The U ₉₅ value was successfully generated.	Generated value
1	The U ₉₅ process generated an uncertainty value exceeding limits (limits to be determined during testing).	-9900
2	The minimum DNI threshold was not met.	-9900
3	The submitted data record was flagged by SERI QC as missing one or more of the three components.	-9900
4	The submitted data record represented a zenith angle greater than the specified maximum.	-9900
5	The submitted data generated a SERI QC flag above the specified maximum.	-9900
6	The system was not able to calculate U _o SYS because of zero or abnormally low Kn + Kd in the denominator.	-9900
7	SERI QC returned an error code (this will halt execution).	-9900

4.3 Assumptions for Roles and Responsibilities

The key activities and suggested roles and responsibilities are summarized in Table 8 using the RACI approach. This accepted project management tool identifies these key levels of involvement of an entity or person for each activity:

- Responsible—performing an activity or does the work
- Accountable—ultimately accountable to the client and has yes/no/veto authority (only one entity or person can be assigned this role). The NREL client is the U.S. Department of Energy, and the Solar Resource Solutions, LLC (SRS) client is NREL.
- Consulted—providing feedback and contributing to the activity
- Informed—requiring knowledge of the decision or action.

Table 8. Summary of Key Activities, Roles, and Responsibilities

Activity	NREL	SRS
System design concepts	C I	R A C I
Final system specifications	R A C I	R C I
System software coding	R A C I	C I
Software testing & validation	R A C I	R C I
QCFIT documentation & distribution	R A C I	C I
Radiometer uncertainty spreadsheet documentation & distribution	R A C I	C I
Sample test case for distribution with the system software	R A C I	C I
System software documentation	R A C I	C I

4.4 Nonfunctional Requirements

The SUNI application is intended to run on a stand-alone computer or on a dedicated partition/account in a multitasking environment. Given the limited interaction with a user base or a networked environment, the nonfunctional requirements are simple and straightforward.

Nonfunctional requirements are minimal but are provided here to clarify the context of the system.

4.4.1 System Description

The software is conceptually simple as an input-process-output model with a user interface provided to facilitate the necessary setup for processing. The software allows a user to present a file of solar irradiance measurements and return a file with the associated quality flags and estimated U_{95} uncertainties for each qualifying measurement.

4.4.2 Development Software

NREL has stipulated that the system will be developed using the Python and JavaScript programming languages.

4.4.3 Modularity

The software will be designed with a modular scheme to not only facilitate easier maintenance but also allow clean extraction of the core routines to interface with another program. In this manner, the entire functionality of the uncertainty process can be abstracted in a simple function call and parameter list. No user interface capability is included in the core uncertainty process routine; it is strictly a called function that returns a set of values and status codes.

4.4.4 Error Detection and Recoverability

The software design will provide error detection to prevent either wrong answers or an uncontrolled failure. Any situation that results in such conditions should cleanly halt processing

before failure and propagate the error messages through the hierarchy to the user interface. Many error conditions are inconsequential to continued program execution but will be important for flagging the uncertainty data. As the software is developed, all error conditions will be identified, and unique codes will be assigned to communicate conditions to the user.

4.4.5 Portability

The software should be created in the Python development environment with an expectation that it will be deployed on Mac OS, Unix/Linux, and Microsoft Windows, with minimal allowances made for any differences relating to file manipulation or GUI construction.

4.4.6 Security

The system is designed to be hosted on a stand-alone computer or terminal and will be subject to the physical and operating system security of the host system. The GUI and file IO are within direct control of the software and are within operating system oversight. Even with, for example, a high-value proprietary input data file, no special security protocols are necessary beyond those provided by the operating system and controlled physical or remote access to the computer.

4.4.7 Performance

The system neither limits the size of the input file nor specifies the processing capabilities of the hardware, and thus a performance timing metric is not applicable; however, all reasonable efforts will be made in the program design for efficient IO, math routines, and other system attributes to enable a reasonably fast processing environment.

4.4.8 Scalability

The design should anticipate enhancements in the future. This could be in the form of additional uncertainty protocols or even accommodating additional types of radiometers.

4.4.9 Testability

A suite of input files and configuration scenarios will be developed to test and benchmark the system. A small set of input files were developed during the proof of concept, and these can serve as an initial test of system performance. Additional data sets from a wider range of data collection scenarios will be developed to serve as a benchmark for future enhancements or implementations within other software.

4.4.10 Accessibility

All user interface designs will comply with the Americans with Disabilities Act of 1990.

4.5 Database Requirements

Several databases will hold the information required to process the data file, produce the SERI QC flags, and assign the uncertainty values. These will be simple text databases. The use of the text-based database format eliminates the need for programming complex database management interfaces with the system. Generally, these files will be in CSV format and will be created by the system to ensure format compliance.

4.5.1 Station Information

The station information will serve to identify a station by name and a short unique identification (ID) code. Through the station ID code, a link can be established to the station's QC-zero files used by SERI QC. The format of the QC-zero file and information it contains is shown in Appendix A. The QC-zero file is created independently from this process by the QCFIT (Maxwell et al. 1993) program.

The station database will contain this information:

- Station identifier (primary key, must be unique)
- Station name (plain text)
- Network name (plain text)
- Instrument IDs for GHI, DNI, and DHI from the most recent processing (to facilitate repeated or ongoing processing without the need for instrument information reentry).

(Note: Station latitude, longitude, and time zone are part of the QC-zero file.)

4.5.2 Instrument Information

Each instrument in use throughout the period of record of the input data file will be included in the instrument database. The instrument database will contain this information:

- Instrument manufacturer
- Instrument model
Instrument serial number (The database key will be a composite of the model and serial number.)
- Calibration history:
 - Calibration date
 - Responsivity
 - Expanded uncertainty (percentage) (from the NREL SOLARUN application).

Note that historical instrument information is not necessary for processing but will serve as documentation of which calibration instance was used for the processing.

This information will be compiled and maintained in an Excel spreadsheet. As new information is added, the spreadsheet will be exported to a .CSV file in the system instrument directory.

4.6 Error Handling

Each status code returned from the subordinate routines will be examined to determine if an interruption in processing is applicable. Errors will be categorized, and the user will be prompted for resolution. Table 9 summarizes the error conditions and required actions.

Table 9. Error Conditions and Resolutions

Error Condition	Action	User Resolution
Input file not found	The program will allow the user to select from existing files, virtually ensuring its presence, but if the file is not present at run time, processing will halt, and the user will be notified.	User can reconfigure to select an existing file, and the processing can be restarted
Required configuration data not present	User is prompted for additional configuration	Processing can restart after proper configuration
Improper data input format	Processing halts and the user is notified of the invalid input format with line number	User likely exits the program and corrects input file format.
Insufficient disk space for output file	Processing halts and the user is notified of insufficient disk space	User exits the program and clears additional disk space, and the program is restarted
SERI QC error (return code other than 0; includes date/time, QC-zero file missing or improper format, undefined Kt/Kn max or undefined boundaries for station)	Processing halts and the decoded SERI QC error is presented to the user	User will likely exit the program and resolve the SERI QC error condition
Uncertainty process fails due to parameters beyond specifications	Mark the uncertainty field as unassignable (-9900), and create the appropriate code for the output data stream.	Processing will continue without notification to the user

5 Software Design Specifications

The software design employs a functional approach with identifiable modules connected by function calls and the passing of abstracted data structures through the function interfaces.

The program will present a user interface that guides the user through a configuration that specifies the station information, uncertainty specifics about the solar radiometers used for the measurements, and the location of the input and output files. Once the configuration information has been entered, the user can initiate the process to read and process all records in the input file, creating an output file holding the calculated uncertainty values and SERI QC flags as fields added to the input records.

5.1 Program Schematic

Once the user initiates the file processing, the records in the input file will be transferred to the uncertainty process one at a time, along with radiometer uncertainty data. Figure 5 shows the hierarchy of processing as the uncertainty process transfers the necessary information in turn to modules for SERI QC, operational uncertainty, and uncertainty merging. Each module returns specific values and a status code. A failure status code will halt the uncertainty process and return control to the user interface, which will interpret the failure mode and present a message to the user.

5.2 Internal Data Structures

Throughout the processing, information will be passed among modules using data structures. Information entered on the main GUI will be retrieved from the various controls and stored in the appropriate data structure. The data structures (some with substructures) are described as follows:

//Structure definitions

```
// Core uncertainty parameters (passed to and from the uncertainty process
// for a single set of measurements. This is the only conduit with the calling
// program.)
Struct uDat
    // input values
    Char  stationID[20]// for SERI QC (includes access to lat, lon, timezone)
    Char  QC0Dir[MAX_PATH_LEN]//path to QC0 file
    int   yr          // data year timestamp
    int   mo          // data month timestamp
    int   dy          // data day timestamp
    int   hr          // data hour timestamp
    int   mn          // data minute timestamp
    int   intrvl      // timestep interval of the measurement data
    float GHI         // GHI measurement
    float DNI         // DNI measurement
    float DHI         // DHI measurement
    float urGHI       // GHI radiometer uncertainty
    float urDNI       // DNI radiometer uncertainty
    float urDHI       // DHI radiometer uncertainty
    float DNImIn      // minimum DNI threshold
    float Zmax        // maximum zenith angle threshold
    int   QCmax       // maximum allowable SERI QC flag
```

```

// return values
int      qcGHI    // SERIQC quality code for GHI
int      qcDNI    // SERIQC quality code for DNI
int      qcDHI    // SERIQC quality code for DHI
float    U95GHI   // expanded GHI uncertainty
float    U95DNI   // expanded DNI uncertainty
float    U95DHI   // expanded DHI uncertainty
float    Usys     // interim system uncertainty
float    UsysAbs  // interim absolute value system uncertainty
float    Ufield   // interim field uncertainty
int      SQCode   // return SERIQC status code
int      uCode    // return uncertainty status code
float    zen      // Zenith angle
float    ETR      // extraterrestrial solar
float    ETRn     // extraterrestrial direct solar
float    kt       // GHI k-space
float    kn       // DNI k-space
float    kd       // DHI k-space

// Top level structure with sub-structures
Struct configDat
    struct station
    struct instruments
    struct files
    struct ret_status
    struct defaults
    struct uDat
    struct statistics
    bool cancel // flag to interrupt main processing loop

// Substructure for station and SERI QC info
struct station
    char ID[20]
    char name[100]
    float latitude
    float longitude
    float timezone
    int

// Substructure for radiometers
Struct rad
    char SN[30]
    char mode[50]
    char measType[4]
    char calDate[11]
    char dueDate[11]
    float rs
    float Ur // radiometer database uncertainties
    float U95 // final measurement uncertainty
    int code // uncertainty result code

// Substructure for the three measuring instruments
Struct instruments
    Struct rad GHI
    Struct rad DNI
    Struct rad DHI

```

```

// Substructure for input and output files
Struct files
    char input_path[maxPathLength]
    char input_fname[maxPathLength]
    FILE infile
    char output_path[maxPathLength]
    char output_fname[maxPathLength]
    char SERIQC_path[maxPathLength]
    FILE outfile
    bool extendedRpt
    bool overwrite_output
    int lineCtr
    time_t   procTime //system time when processing starts
    char begTime[20] // timestamp of first data record
    char endTime[20] // timestamp of last data record

// Substructure for return status and error codes
Struct retStatus
    int SERIQC
    int instUncert
    int opUncert
    int mergeUncert

// Substructure for defaults
Struct defaults
    int max_SERIQC
    float min_DNI
    float max_zen

// Substructure stats for individual instruments
Struct stats
    double sum // sum of values for stdev
    double sumSq // sum of the values squared for stdev
    double mean // calculated mean
    double stdev // calculated standard deviation

// Substructure for statistics for all instruments
Struct statistics
    float Usys
    int nValid
    float Sysn // System uncertainty counter
    float SysSum // System uncertainty sum
    float Ufn // Field uncertainty counter
    float Ufsum // Field uncertainty sum
    float Urn // Urads uncertainty counter
    float Ursum // Urads uncertainty sum
    Struct stats GHI
    Struct stats DNI
    Struct stats DHI
    errorCount[ERROR_LIST_SIZE] // dimension by number of status codes defined

```

5.3 User Interface

To preserve the fundamental functionality of the core uncertainty processing, all file IO and user interactions are segregated in a top-level function. This function will call the uncertainty process

on a record-by-record basis and receive in return the final U_{95} uncertainty value of a single record. This design allows the uncertainty process to be extracted and incorporated with other applications that require a different mode of operation or accommodate different file formats.

5.3.1 Interface and Program Initialization

At program startup, several structures and controls will be initialized.

Main function

```
// local variables
char instrument_string[300] // stores cal and uncertainty for an
instrument
char SN[30] // temporary serial number
char model[50] // temporary model holder
char type[4] // temporary measurement type
char label[80] // temporary combo of SN and model

// initialize SERIQC station list
Change internal directory to files.SERIQC_path
Get listing and count of the directory using wildcard spec "*.qc0"
Loop [i] through qc0file_count
    Open(qc0file[i])
    Read first line and parse station_id and station_name
    Concatenate and populate control_list[i] pulldown

// initialize instrument pulldowns
Initialize array of inst_strings
Change internal directory to the instrument uncertainty directory
Open the Instrument Calibration Data file
Loop I through each file record
    Copy complete record as CSV string to instrument_string[i]
Sort instrument_string array by SN //this will sort by serial number

Loop i through the sorted instrument_string array
    Parse SN, model, and type from instruments_string[i]
    Concatenate SN and model (with space) to the label string
    Case (type)
        GHI: Copy label to GHI_control list; note array index
        DHI: Copy label to DHI_control list; note array index
        DHI: Copy label to DHI_control list; note array index

// retrieve last GUI configuration and initialize booleans
Change internal directory to .cfg folder
Open file for reading "LastCfgSave.cfg"
Populate main UI controls with corresponding values
configDat.cancel = FALSE

// display and activate main User interface
End
```

5.3.2 Main User Interface

A conceptual view of the user interface screen is described next. The panel should be implemented in a conservative motif with an uncluttered layout and without bright colors. Use of a highlight color is acceptable to indicate error messages.

At program startup, a user interface will be presented (Figure 6), prompting the user to configure a data processing session. All of these fields can be saved in a retrievable configuration to facilitate easy setup for a future session. Configurations can be saved with a unique name to accommodate different setups, e.g., for different stations with different instruments.

- At program close, the current configuration will automatically be saved and will populate the GUI at the next program startup.
- The current configuration can be saved with a user-specified name through the main user interface menu *File*→*Save Configuration*.
- Previously saved configurations can be accessed through the main user interface menu *File*→*Open Configuration*.

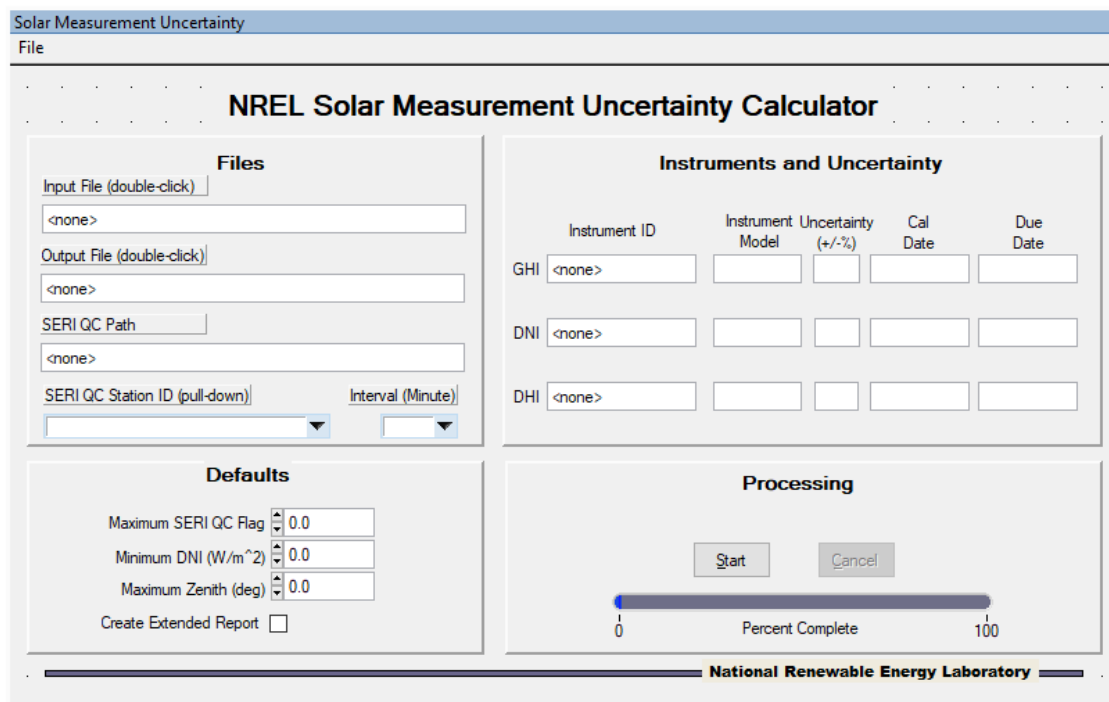


Figure 6. Main GUI

The functionality and pseudocode for each interface control is described as follows:

Input file—Double-clicking on the *Input File* control will invoke a file browser allowing the user to browse to a desired folder and select a single file that contains the measurements to be evaluated. The interface will have a cancel option to return to the main user interface. The path and name of the file will be retained in the data structure and displayed on the main user interface.

Function Input_File_clicked (parameter: struct files)

```
Set browser default directory = files.input_path // previously set
Set browser to allow directory changes
```

CALL browser

```
If browser.cancel
    Close browser
```

```
Files.input_fname = browser selection
Parse into files.input_path and files.input_fname // for report
```

```
Close browser
```

```
Set the input file control.text to files.input_fname  
End
```

Output file—Double-clicking on the *Output File* control will invoke a file browser allowing the user to select or create a folder and file name to create a file to hold the output data from the process. The following functionality will be implemented when specifying an output file name:

- Check for an existing file of the given name and present an assertive yes/no popup to overwrite the file. If yes, processing will continue; if no, the field will be blanked and wait for another input.
- Check for correct file naming conventions. Given different naming conventions among the possible operating systems for deployment, this can most easily be accomplished by opening a file with the given name. If no open error occurs, the file path and name will be accepted and stored in the data structure and displayed on the main user interface. A file successfully opened will be closed and deleted before proceeding.
- If the file naming convention test fails, a popup file name error will be presented, and the user will acknowledge and then be given the opportunity to attempt another name.
- At the file browser, a cancel option will be provided to return control to the main user interface without specifying a name.

```
Function Output_File_clickevent (parameter: struct files)  
Set browser default directory = files.output_path //previously set  
Set browser to allow directory changes  
CALL browser  
If browser.cancel  
Close browser  
Files.output_fname = browser selection  
If (Openfile(files.output_fname, as read)) == existing  
Popup_error("File exists - Overwrite (Y/N?")  
If popup return = N  
Close browser  
ElseIf (Openfile(files.output_fname, as write)) == fail  
Popup_error("File cannot be created. Try again."  
Else // good file name  
Set output file files.output_fname to the control text  
Parse into files.output_path and files.output_fname // for report  
Delete(output file) // will recreate for processing later  
Close browser
```

```
End
```

SERI QC path—This specifies the path to the user’s SERI QC directory that holds the .QC0 files.

```
Function SERIQC_Path_clickevent (parameter: struct files)  
Set folder_browser default directory = files.SERIQC_path //previously set  
Set folder_browser to allow directory changes  
CALL browser  
If folder_browser.cancel  
Close folder_browser
```

```

Files.SERIQC_path = browser selection

Set SERI QC path files.SERIQC_path to the control text

// update Station control population with same code from program startup
// initialize SERIQC station list
Change internal directory to files.SERIQC_path
Get listing and count of the directory using wildcard spec "*.qc0"
Loop [i] through qc0file_count
    Open(qc0file[i]
        Read first line and parse station_id and station_name
        Concatenate and populate control_list[i] pulldown

```

End

SERI QC station ID—All station information is contained in the SERI QC .QC0 files (see Appendix A), and all such files are created externally to this program using the NREL QCFIT application. The *SERI QC Station* control provides a pull-down list (or other appropriate control) of stations, which was initialized at program startup. Here, the user selects from the list of stations in the control pull-down menu.

```

Function Station_ID_clickevent (parameters: struct station)
    // clickevent generated as user selects from the station ID pulldown list
    Parse station ID and name and assign:
    Station.ID      // used for SERI QC calls
    Station.name    // used for reports

```

Interval (minute)—This is the time step interval (or integration time) of the data in the input file, in minutes. This value is limited by SERI QC to be between 1 and 60 minutes inclusive. The control will limit the selected value to that range.

```

Function Interval_clickevent (parameters: struct station)
    // click event generated by entering or selecting an interval value
    Station.interval = control value

```

Instruments and uncertainty—A flat-file database will exist in the instrument directory that holds information for all instruments eligible for inclusion in the uncertainty processing. See Section 5.8.2 for a complete description. The CSV file will hold the information in Table 20.

This file will be read by the system during program initialization and used to populate the pull-down list control for the instrument selection. From this table, only the *Instrument Expanded Uncertainty* will be used by the uncertainty process; however, uncertainty report files will include all instrument information to document the instrument deployments at the time of the measurements.

```

Function GHI_list clickevent (parameter: struct instruments)
    indx = GHI_list_index;
    Parse instrument_string[indx] to SN, Model, Ur, CalDate, DueDate
    Populate controls for ID, Model, uncertainty, Cal Date and Due Date
    // save values to instrument structure
    instruments.GHI.Ur = parsed Ur
    instruments.GHI.calDate = parsed calDate

```

```
instruments.GHI.dueDate = parsed dueDate
End
```

```
Function DNI_list clickevent (parameter: struct instruments)
  indx = DNI_list_index;
  Parse instrument_string[indx] to SN, Model, Ur, CalDate, DueDate
  Populate controls for ID, Model, uncertainty, Cal Date and Due Date
  // save values to instrument structure
  instruments.DNI.Ur = parsed Ur
  instruments.DNI.calDate = parsed calDate
  instruments.DNI.dueDate = parsed dueDate
End
```

```
Function DHI_list clickevent (parameter: struct instruments)
  indx = DHI_list_index;
  Parse instrument_string[indx] to SN, Model, Ur, CalDate, DueDate
  Populate controls for ID, Model, uncertainty, Cal Date and Due Date
  // save values to instrument structure
  instruments.DHI.Ur = parsed Ur
  instruments.DHI.calDate = parsed calDate
  instruments.DHI.dueDate = parsed dueDate
End
```

Defaults—The system allows the user to modify certain system defaults to accommodate processing considerations unique to a particular data set. Three default overrides are available:

- **Maximum SERI QC flag**—The system default includes all three-component SERI QC flags through 87 (see the SERI QC flags description in Section 4.2.2), beyond which the data are considered egregious and outside a threshold to justify inclusion. The Flag 87 limit is somewhat arbitrary based on SERI QC conventions, beyond which data fall into a region of unquantified error. For a more conservative analysis, users have the option to lower the error threshold for inclusion in processing. For consistency in filtering, the maximum flag should be restricted to the following values: Flag 3 or 9 or 13, 17, 21,...87 [i.e., 13–87 in increments of 4].

```
Function max_SERIQC clickevent(parameter: struct defaults)
  Defaults.max_SERIQC = control value
End
```

- **Minimum DNI**—This is the lowest value for DNI allowed in a measurement record to be eligible for processing. The minimum is needed to filter out small values that could cause unwanted large ratios between small numbers. The default is 25 W/m². It is plausible in certain measurement scenarios that the value could be lowered to include more data. Other scenarios might require a higher number if unrealistic uncertainty values occur.

```
Function minimum_DNI clickevent(parameter: struct defaults)
  Defaults.min_DNI = control value
End
```

- **Maximum zenith angle**—At high zenith angles (with the sun close to the horizon), certain instrument characteristics can cause large relative errors, resulting in unrealistic

uncertainties. This value can be changed based on knowledge of the instruments or unwanted processing behavior.

```
Function minimum_zen clickevent(parameter: struct defaults)
    Defaults.max_zen = control value
End
```

- Create extended report—This option will provide values for the interim system uncertainty (UoSYS) and field uncertainty (UoField) in the output data file and the summary statistical reports.

```
Function extendedReport clickevent(parameter: struct defaults)
    configDat.files.extendedRpt = control value
```

Start processing—When the configuration is complete and validated, the user will initiate processing with the *Start* control on the user interface. The *Start* control will dim, and the *Cancel* control will be active. All other main GUI controls except for the progress bar will be inactive.

See Section 5.4 for pseudocode.

Cancel processing—During processing, the user can abort the processing by clicking on the *Cancel* control on the user interface. The *Start* control will return to the active state, and the *Cancel* control will dim. All other main GUI controls will be reactivated.

See Section 5.4 for pseudocode.

File menu—The file menu presents three options:

Save configuration—At the completion of filling all fields in the main user interface, the configuration can be saved to a user-specified configuration file. The contents of the configuration file are detailed in Section 5.7.3. From the File pull-down menu, select:

File→*Save Configuration*

This brings up a dialog window prompting the user to enter the desired name of the file. The configuration will be saved in the system's .\cfg directory. If a configuration with the specified name already exists, an assertive yes/no overwrite popup will be displayed. If yes, the configuration will be saved by that name; if no, the field will be blanked and wait for another input.

The function will provide a *Cancel* option.

```
Function Save_config menu click event (parameter: configDat)
    Open file browser and set internal directory to cfg folder
    Disallow folder changes
    Open the user-specified file for writing
    Save all configurations
    Close file
```

End

- Retrieve a configuration—To retrieve previously saved configurations (detailed in Section 5.7.3), from the file pull-down menu, select:

File→*Get Configuration*

This brings up a dialog window with a pull-down list of all previously saved configurations. When selecting a configuration, all fields in the main user interface will be populated with the values that existed when the configuration was saved.

This function will provide a *Cancel* option.

```
Function Get_config menu clickevent(parameter: configDat)
  Open file browser and set directory to cfg folder
  Disallow folder changes
  Open user-selected file for reading
  Read all configurations and populate GUI and structures
  Close file
End
```

- Close program—This option will dismiss the GUI and close the program. This code will also be executed when clicking on the GUI's red close-window X.

```
Function CloseProgram menu clickevent(parameter: configDat)
  Change directory to cfg folder
  Open file "LastCfgSave.cfg" for writing
  Write all configurations
  Close file
  Dismiss main GUI
  End program
End
```

5.4 Processing the Data

When the processing configuration is complete and validated, the user will initiate the processing with the *Start* control on the user interface, and the click event will pass control to the processing subroutine, as described here.

5.4.1 IO Loop

The user interface will be suspended (except for the *Cancel* button), and the program control will flow to the main `IO_process` function, which will open the input file and loop through each record, submitting each to the uncertainty process, receiving the return values, and writing the results to the output file. The uncertainty process is designed to be a stand-alone function that can be extracted and inserted into other program code.

```
Function start_control clickevent(parameter: configDat)
// at user's process command, transfer UI control values to data structures
// IF they have not already been assigned by a click event or initialization.
  Get configDat.station.stationID from Interface
```

```

Get configDat.files.SERIOQpath from Interface
Get configDat.files.input_fname from Interface
Get configDat.files.output_fname from Interface
Get configDat.instruments.GHI.SN from Interface
Get configDat.instruments.DNI.SN from Interface
Get configDat.instruments.DHI.SN from Interface
Get configDat.instruments.GHI.Ur from Interface
Get configDat.instruments.DNI.Ur from Interface
Get configDat.instruments.DHI.Ur from Interface
Get configDat.instruments.GHI.calDate from Interface
Get configDat.instruments.DNI.calDate from Interface
Get configDat.instruments.DHI.calDate from Interface
Get configDat.files.extendedRpt from interface
Configdat.files.startTime = [system time]

// begin processing
CALL Main_IO_Process (configDat)

// Return control to the UI

Int Function Main_IO_Process (parameter: struct configDat)

    // Define status and error codes (these should be in a header/include file
for the stand-alone uncertainty function)
STATUS_VALID = 0
ERROR_SERIOQ = 1
ERROR_3_COMP = 2
ERROR_QCMAX = 4
ERROR_HI_Z = 5
ERROR_LO_DNI = 6
ERROR_ETR = 7
ERROR_KSPACE = 8
ERROR_HI_UNCERT= 9
ERROR_LIST_SIZE= 10 // defines the size of zero-based error code list

// output file header definitions
// standard
Char headerS =
    "Date (MM/DD/YYYY),Time (HH:MM)," // def continues 6 lines w/o wrap
    "GHI (W/m^2),GHI SERI QC Flag,"
    "GHI Uncertainty (+/-%),GHI Uncertainty Code,"
    "DNI (W/m^2),DNI SERI QC Flag,"
    "DNI Uncertainty (+/-%),DNI Uncertainty Code,"
    "DHI (W/m^2),DHI SERI QC Flag,"
    "DHI Uncertainty (+/-%),DHI Uncertainty Code\n"

// extended
Char headerE =
    "Date (MM/DD/YYYY),Time (HH:MM)," // def continues 7 lines w/o wrap
    "GHI (W/m^2),GHI SERI QC Flag,"
    "GHI Uncertainty (+/-%),GHI Uncertainty Code,"
    "DNI (W/m^2),DNI SERI QC Flag,"
    "DNI Uncertainty (+/-%),DNI Uncertainty Code,"
    "DHI (W/m^2),DHI SERI QC Flag,"
    "DHI Uncertainty (+/-%),DHI Uncertainty Code,"

```

```

    "Field Uncertainty (+/-%),System Uncertainty (+/-%)\n"

// Zero all accumulators and counters in the configDat.statistics structure
configDat.statistics.n          = 0

configDat.statistics.GHI.sum    = 0.0
configDat.statistics.DNI.sum    = 0.0
configDat.statistics.DHI.sum    = 0.0

configDat.statistics.Sysn      = 0
configDat.statistics.SysSum    = 0.0
configDat.statistics.uFn       = 0
configDat.statistics.UFSum     = 0.0
configDat.statistics.uRn       = 0
configDat.statistics.URSum     = 0.0

// zero the status code counters
loop i (0 to <ERROR_LIST_SIZE)
    configDat.statistics.errorCount[i] = 0

// NULL the beginning and ending timestamps
configDat.files.begTime = NULL
configDat.files.endTime = NULL
// open input and output files
configDat.files.infile = open(configDat.files.input_fname)
If open fails
    Popup_OK("Cannot open input file [name])
    Return

configDat.files.outfile = open(configDat.files.output_fname)
If open fails
    Popup_OK("Cannot open output file [name])
    Return

// print output file header
If (extended box checked)
    Print(outfile, headerE)
Else
    Print(outfile, headerS)

configDat.files.lineCtr = 0
fileSize = (Call function to get the Input_file_size)
recSize = Priming read of inputRecord from inFile
While not EOF(configDat.inFile)
    bytesRead += recSize
    pctRead = bytesRead / fileSize * 100
    update progress bar with pctRead

    Parse inputRecord for date, time, irradiances; move to uDat structure
    Check for proper field count during parse
    If field count fails
        PopupYN("Input file format error in line %i. Continue?",
            configDat.lineCtr)
        If No
            Close input and output files
            Return

```

```

If configDat.files.begTime = NULL
    Copy record date and time to begTime

configDat.files.lineCtr++
retCode = CALL Uprocess (uDat)
if (retCode=ERROR_SERIQC)
    ErrorMessage = SERIQC_Decode(configDat.retStat.SERIQC)
    Strcat("Line #[linectr]: SERI QC Error", ErrorMessage, ". Processing
will halt")
    Popup_OK(ErrorMessage)
    Close infile
    Close outfile
    Return ERROR_SERIQC // abort with fatal error

// increment status counters
configDat.statistics.errorCount[retCode]++

// save the return code for writing with the data
uDat.uCode = retCode

// if uncertainty is valid, update stats
If (retCode = STATUS_VALID) // good uncertainty calculations
    // save results and increment counters
    n = configDat.statistics.nValid // just a shorter array subscript

    configDat.statistics.GHI.values[n] = uDat.GHI.U95
    configDat.statistics.GHI.sum += uDat.GHI.U95
    configDat.statistics.GHI.sumSq += (uDat.GHI.U95)^2

    configDat.statistics.DNI.values[n] = uDat.DNI.U95
    configDat.statistics.DNI.sum += uDat.DNI.U95
    configDat.statistics.DNI.sumSq += (uDat.DNI.U95)^2

    configDat.statistics.DHI.values[n] = uDat.DHI.U95
    configDat.statistics.DHI.sum += uDat.DHI.U95
    configDat.statistics.DHI.sumSq += (uDat.DHI.U95)^2

    configDat.statistics.Sysn++
    configDat.statistics.SysSum += configdat.statistics.Sys

    configDat.statistics.uFn++
    configDat.statistics.uFsum += configdat.statistics.Ufield

    configDat.statistics.uRn++
    configDat.statistics.uRsum += configdat.statistics.Urads

    configDat.statistics.nValid++
End if good code

retVal = CALL output(uDat, configDat.files)
if (retVal <> STATUS_VALID)
    PopupOK("Unable to write to the output file for line %i\n"
        ". Process stopped.", configDat.files.lineCtr++)
    Close(configDat.files.outfile)
    Close)configDat.files.infile)
    // other cleanup as necessary.
    Return

```

```

    recSize = read_input_record from infile
End at EOF

Copy end date and time to configDat.files.endTime

// Generate summary statistics
CALL doStats(configDat.statistics)
retVal = CALL doReports(configDat)

```

Return

5.4.2 Output

Once an input record has been processed, the results will be written to a line in the output file (see Section 5.6.2 for a sample output file record).

```

Int FUNCTION output(struct uDat, struct files)
    Char outfmt1 = "02i/02i/04i,02i:02i,0.1f,02i,0.1f,2i," // continues
        "0.1f,02i,0.1f,2i,0.1f,02i,0.1f,2i"
    Char outfmt2 = ",0.1f,0.1f"
    retVal = Print(files.outfile, outfmt1,
        uDat.mo,
        uDat.dy,
        uDat.yr,
        uDat.hr,
        uDat.mn,
        uDat.GHI,
        uDat.qcGHI,
        uDat.U95GHI,
        uDat.uCode,
        uDat.DNI,
        uDat.qcDNI,
        uDat.U95DNI,
        uDat.uCode,
        uDat.DHI,
        uDat.qcDHI,
        uDat.U95DHI,
        uDat.uCode)

    If retVal shows write failure
        Return (ERROR_WRITE_FAIL)

    If (files.extendedRpt)
        retVal = Print(files.output, outfmt2, measurements.Usys,
            measurements.Ufield)
    If retVal shows write failure
        Return (ERROR_WRITE_FAIL)

    Print(files.output, "\\n")

    Return (STATUS_VALID)

```

5.4.3 Summary Statistics

This section describes the function to calculate the means and standard deviation (GHI shown; duplicate for DNI and DHI).

Function doStats(parameter: struct statistics)

```
// calculate means
If (statistics.n > 0)
    GHI.mean = GHI.sum / statistics.n
Else
    GHI.mean = -9900.0

// calculate standard deviations
sumDiff = 0.0
If (statistics.n > 1)
    GHI.stdev =
        sqrt(GHI.sumSq - GHI.sum^2/statistics.n) / (statistics.n-- 1)
Else
    GHI.stdev = -9900

// (same for statistics.stats.DNI)
// (same for statistics.stats.DHI)
End
```

5.4.4 Summary Reports

At the end of processing, a popup summary statistical report will be generated, and an ASCII report file will be generated with more detail. See Section 5.7 for the format and contents of the summary reports.

Function doReports(parameters: struct configDat)

```
Char rptMsg[2000]
Char tmpStr[500]

totalRecords = configDat.files.lineCtr

// file report
// open report file for input to test existence
fName = ("%sReport.txt", configDat.files.input_fname)
open rptFile with fName as read text
if (open successful) // file exists
    response = PopupYN("File %s exists. Overwrite?")
    if (response = No)
        continue to popup report
Else // open report file for writing
    Open rptFile with fName as write text
    if (open fail)
        response = CALL Popup_YN("Cannot open %s. Try again?", fName)
        If response = 'N'
            Continue to popup report

// line 1
Print(rptFile, "Uncertainty Processing Report for %s\n\n",
    configDat.files.input_fname)

// line 2
Print(rptFile, "Processing Date: %s\n\n", configDat.files.startTime)

// lines 3 and (blank) 4
Print(rptFile, "From %s to %s (%i-minute interval)\n\n",
```

```

        configDat.files.begTime,
        configDat.files.endTime,
        configDat.station.interval)

// line 5
Print(rptFile, "System Configuration:\n"

// lines 6-8
Print(rptFile, "GHI:s/n %s | RS: %0.2fuV/W/m^2 | U95: +/-%0.3f | Cal Date:
%s\n"
        configDat.instruments.GHI.SN,
        configDat.instruments.GHI.rs,
        configDat.instruments.GHI.Ur,
        configDat.instruments.GHI.calDate)

Print(rptFile, "DNI:s/n %s | RS: %0.2fuV/W/m^2 | U95: +/-%0.3f | Cal Date:
%s\n"
        configDat.instruments.DNI.SN,
        configDat.instruments.DNI.rs,
        configDat.instruments.DNI.Ur,
        configDat.instruments.DNI.calDate)

Print(rptFile, "DHI:s/n %s | RS: %0.2fuV/W/m^2 | U95: +/-%0.3f | Cal Date:
%s\n"
        configDat.instruments.DHI.SN,
        configDat.instruments.DHI.rs,
        configDat.instruments.DHI.Ur,
        configDat.instruments.DHI.calDate)

// lines 9-12 (with blank)
Print(rptFile, "SERI QC Max: %i\n", configDat.defaults.max_SERIQC)
Print(rptFile, "Zenith Angle Max: %0.0f\n", configDat.defaults.max_zen)
Print(rptFile, "DNI Minimum: %0.0f\n\n", configDat.defaults.min_DNI)

// line 13
Print(rptFile, "Input Data Records: %i\n", totalRecords)

// line 14
Print(rptFile, "Three-component Records: %i (%0.1f%%)\n",
        // need to determine this source

// line 15
Print(rptFile, "Above SERIQC Max: %i (%0.1f%%)\n",
        configDat.statistics.errorCount[ERROR_QCMAX],
        configDat.statistics.errorCount[ERROR_QCMAX] / totalRecords)

// line 16
Print(rptFile, "Above Zenith Angle Max: %i (%0.1f%%)\n",
        configDat.statistics.errorCount[ERROR_HI_Z],
        configDat.statistics.errorCount[ERROR_HI_Z] / totalRecords)

// line 17
Print(rptFile, "Below DNI Minimum: %i (%0.1f%%)\n",
        configDat.statistics.errorCount[ERROR_LO_DNI],
        configDat.statistics.errorCount[ERROR_LO_DNI] / totalRecords)

// line 18

```

```

// lump ETR and unlikely KSPACE errors together
Errors = configDat.statistics.errorCount[ERROR_ETR] +
        configDat.statistics.errorCount[ERROR_KSPACE]
Print(rptFile, "Mathematically Invalid: %i (%0.1f%%)\n",
      Errors, Errors / totalRecords)

// line 19 and blank 20
Print(rptFile, "Total Eligible Uncertainty Records: %i (%0.1f%%)\n\n",
      configDat.statistics.nValid,
      configDat.statistics.nValid / totalRecords)

// lines 21-23
Print(rptFile, "GHI Mean U95: +/-%0.1f%% | Standard Deviation: %0.1f\n",
      configDat.instruments.GHI.U95,
      configDat.statistics.GHI.stdev)

Print(rptFile, "DNI Mean U95: +/-%0.1% | Standard Deviation: %0.1f\n",
      configDat.instruments.DNI.U95,
      configDat.statistics.DNI.stdev)

Print(rptFile, "DHI Mean U95: +/-%0.1f%% | Standard Deviation: %0.1f\n",
      configDat.instruments.DHI.U95,
      configDat.statistics.DHI.stdev)

// extended report
If (configDat.files.extendedRpt)
  // Line 24 (preceding blank) and 25-26
  Print(rptFile, "\nUrads Uncertainty Mean: +/-%0.2f%%\n",
        configDat.statistics.uRsum / configDat.statistics.uRn)

  Print(rptFile, "\System Uncertainty Mean: +/-%0.2f%%\n",
        configDat.statistics.SysSum / configDat.statistics.Sysn)

  Print(rptFile, "Field Uncertainty Mean: +/-%0.2f%%\n",
        configDat.statistics.uFsum / configDat.statistics.uFn)

// Report completed
Close rptFile

+++++
// popup summary
// line 1 & 2 (blank)
Strprn(tmpStr, "Uncertainty Processing Report for %s\n\n",
      configDat.files.input_fname)
Copy tmpStr to rptMsg

// line 3
Strprn(tmpStr, "Beg: %s. End %s. Data Records: %i\n",
      configDat.files.begTime, configDat.files.endTime, totalRecords)
Concatenate(rptMsg, tmpStr)

// line 4
Strprn(tmpStr, "Total Eligible Records: %i (%0.1f%%)\n",
      configDat.statistics.nValid,
      configDat.statistics.nValid / totalRecords)
Concatenate(rptMsg, tmpStr)

```

```

// line 5 and 6 (blank)
Strprn(tmpStr, "Exceeded SERIQC Max: %i (%0.1f%%)\n\n",
    configDat.statistics.errorCount[ERROR_QCMAX],
    configDat.statistics.errorCount[ERROR_QCMAX] / totalRecords)
Concatenate(rptMsg, tmpStr)

// line 7
Strprn(tmpStr, "GHI Mean U95: +/-%0.1f%% | Standard Deviation: %0.1f\n",
    configDat.instruments.GHI.U95,
    configDat.statistics.GHI.stdev)
Concatenate(rptMsg, tmpStr)

// line 8
Strprn(tmpStr, "DNI Mean U95: +/-%0.1f%% | Standard Deviation: %0.1f\n",
    configDat.instruments.DNI.U95,
    configDat.statistics.DNI.stdev)
Concatenate(rptMsg, tmpStr)

// line 9
Strprn(tmpStr, "DHI Mean U95: +/-%0.1f%% | Standard Deviation: %0.1f\n",
    configDat.instruments.DHI.U95,
    configDat.statistics.DHI.stdev)
Concatenate(rptMsg, tmpStr)

// extended report
If (configDat.files.extendedRpt)
    // Line 10 (preceding blank) and 11
    Strprn(tmpStr, "\nUrads Uncertainty Mean: +/-%0.2f%%\n",
        configDat.statistics.uRsum / configDat.statistics.uRn)
    Concatenate(rptMsg, tmpStr)

    Strprn(tmpStr, "System Uncertainty Mean: +/-%0.2f%%\n",
        configDat.statistics.SysSum / configDat.statistics.Sysn)
    Concatenate(rptMsg, tmpStr)

    // Line 12
    Strprn(tmpStr, "Field Uncertainty Mean: +/-%0.2f%%\n",
        configDat.statistics.uFsum / configDat.statistics.uFn)
    Concatenate(rptMsg, tmpStr)

// report string is complete, print to popup
CALL PopupOK(rptMsg)
End

```

5.4.5 Uncertainty Processing

The uncertainty function (`Uprocess()`) is called from the IO loop with the `uDat` structure, which includes a single record of input data, along with necessary configuration information. The function calls in turn the SERI QC and uncertainty merging functions.

The uncertainty function is designed as a stand-alone unit to allow extraction from this code for insertion into other applications requiring an uncertainty evaluation of data. All function input and output parameters are defined in the `uDat` structure, which must be declared by a header file in the calling routine.

The function returns a status code, which will guide the calling routine in handling errors. Codes are further elaborated in Section 5.5.3.

```

Int Function Uprocess (parameters: struct uDat)
{
    // local variables
    Float Usys // system uncertainty

    // do SERIQC checks
    uDat.SQCcode = CALL seri_qc1(
        uDat.StationID, uDat.QC0Dir,
        uDat.yr, uDat.mo, uDat.dy, uDat.hr, uDat.mn,
        uDat.intrvl, uDat.GHI, uDat.DNI, uDat.DHI,
        uDat.qcGHI, uDat.qcDNI, uDat.qcDHI,
        uDat.ETR, uDat.ETRn, uDat.zen)

    // check for fatal SERI QC error
    If(uDat.SQCcode <> 0)
        Return ERROR_SERIQC // don't process record

    // assure 3 component flag
    If (uDat.qcGHI <> 3 or 9 or (uDat.qcGHI+2)%4 > 1) // not 3 component
        Return ERROR_3_COMP

    // do max SERI QC flag
    If (uDat.qcGHI > uDat.qCmax // don't process record
        Return ERROR_QCMAX

    // do zenith angle max
    If (uDat.zen > uDat.Zmax)
        Return ERROR_HIGH_Z // don't process record

    // check for low limits on DNI
    If (uDat.DNI < uDat.DNImin)
        Return ERROR_LO_DNI // don't process record

    // calculate kspace
    If (uDat.ETR and uDat.ETRn > 0) // don't divide by zero
        uDat.kt = uDat.GHI / uDat.ETR
        uDat.kn = uDat.DNI / uDat.ETRn
        uDat.kd = uDat.DHI / uDat.ETR
    else
        return ERROR_ETR // don't process record

    // get system uncertainty
    If (uDat.kn + uDat.kd) > 0
        uDat.Usys = ((uDat.kt / (uDat.kn + uDat.kd))-1) *100
        uDat.UsysAbs = abs(uDat.Usys)
        if (uDat.UsysAbs > uDat.UoSYSmax
            Return ERROR_HI_UNCERT
    Else
        Return ERROR_KSPACE
    Capture error if there is any possibility of bad data at this point (TBD)

    // merge uncertainties
    retVal = CALL Umerge(uDat)
}

```

```

Return retVal
End

```

5.4.6 SERI QC

SERI QC will be modified to return the internally calculated values for ETR, ETRn, and solar zenith angle. These values should be returned in an expanded function parameter list from these lines in the existing SERIQC.C code:

```

Line#      Code
855      Solzen = Solzen / Sunup;
856      ETR      = ETR      / (float) Intrvl;
857      ETRN     = ETRN     / (float) Intrvl;

int FUNCTION seri_qc1 (
char *Site,      /* S_<Site>.QC0, where <Site> is the site identifier. */
char *QC0Dir,   /* directory where QC0 files are installed */
int Iyear,      /* The year, e.g., 1988 */
int Month,      /* The month of the year (1-12) */
int Iday,       /* The day of the month (1-31) */
int Ihour,      /* The hour of the day (0-24) */
int Minute,     /* The minute of the hour (0-59) */
int Intrvl,     /* The averaging interval in minutes (1-60) */
double Global, /* Global horizontal broadband solar radiation, W/sq m */
double Direct, /* Direct normal broadband solar radiation, W/sq m */
double Difuse, /* Diffuse horizontal broadband solar radiation, W/sq m*/
int *IQCglo,    /* The GLOBAL quality control flag */
int *IQCdir,    /* The DIRECT quality control flag */
int *IQCdif     /* The DIFUSE quality control flag */
float *ETR      /* Extraterrestrial, horizontal, adjusted for partial
                interval */
float *ETRn     /* Extraterrestrial normal to beam, adjusted for partial
                interval */
float *SolZen   /* Solar Zenith Angle adjusted for partial interval */

```

```
RETURN
```

SERI QC return code—Each condition is represented by one bit in the integer return code as described in Table 10 (this could be 16 bits or 32 bits, depending on the platform, but only the first 13 bits are used). When an error condition is encountered, the appropriate bit is set. Multiple bits can be set if more than one error condition is encountered. After the call to SERI QC, the calling program should examine the return code. If it is zero (no bits set), the other return parameters are considered valid. If the return code is nonzero, it can be decoded bit by bit to determine which condition(s) caused the function to terminate.

Table 10. SERI QC Return Codes

Bit	Parameter	Range or Error Condition
0	Site name	Missing
1	Month	1–12
2	Day	1–31
3	Hour	0–24

Bit	Parameter	Range or Error Condition
4	Minute	0–59
5	Composite time	00:00:00–24:00:00
6	Interval	1–60
7	QC-ZERO file	Missing
8	QC-ZERO format	File format corrupt
9	QC-ZERO right boundary	Boundary undefined
10	QC-ZERO left boundary	Boundary undefined
11	QC-ZERO Kt max	Boundary undefined
12	QC-ZERO Kn max	Boundary undefined

5.4.7 Uncertainty Merging

Prior to uncertainty merging, several components of the uncertainty process have been determined (individual radiometer uncertainties, U_{oSYS} , and K-space values). This function combines these values for a final U_{95} for each measurement.

```

Int FUNCTION Umerge(struct uDat)
// local variables
float kSum      // sum of total
float kFrac     // fraction of total
float UdhiFrac // DNI proportion of DNI+DHI
float UdniFrac // DHI proportion of DNI+DHI
float uRads    // total radiometer uncertainty

// calculate fractional uncertainty for DIR and DHI
kSum = unDat.kn + uDat.kd
if (kSum > 0)
    kFrac = uDat.kn / kSum
    UdniFrac = uDat.Urdni * kFrac
    kFrac = uDat.kd / kSum
    UdhiFrac = uDat.Urdhi * kFrac
Else
    Return ERROR_KSPACE

// calculate total radiometer uncertainty (see Equation 6 for clarity)
uRads = 2 *
    sqrt((uDat.urGHI/2)^2 +
        (((UdniFrac) + (UdhiFrac))/sqrt(3))^2)

// remove combined instrument uncertainty from Usys for residual field Uncert
uDat.Ufield = MAX(uDat.UsysAbs - Urads, 0) // MAX prevents negative

// Calculate final U95 values as percent
uDat.U95GHI = sqrt((uDat.urGHI/2)^2 + (uDat.Ufield/2)^2) * 2

uDat.U95DNI = sqrt((uDat.urDNI/2)^2 + (uDat.Ufield/2)^2) * 2

uDat.U95DHI = sqrt((uDat.urDHI/2)^2 + (uDat.Ufield/2)^2) * 2

```

```
Return STATUS_VALID // Here I assume nothing can go wrong in this
function. True?
End
```

5.5 Error Handling

Error handling design will segregate errors into three functionalities that correspond with the three major components of the system:

- GUI errors
- IO errors
- Uncertainty process errors.

5.5.1 GUI Errors

Each component of the GUI will be checked for completeness and correctness prior to allowing control to pass to the IO module.

5.5.2 IO Errors

When control is passed for IO processing, various file errors can occur when opening, reading, or writing files. Any return codes from the operating system should be captured, interpreted in the context of the processing, and return a message to the user that processing will halt. Not all fails might be available from the operating system; in that case, messages will need to be more generic.

All IO errors will result in a stop to processing because there is a high likelihood that corrupt data will otherwise pass to the uncertainty process. Some might allow a user to try again (e.g., if the file is locked by another process, the user can free the file and try again).

Cannot open the input or output file:

- Invalid file name
- File locked by another user
- Too many files already open
- Insufficient memory to allocate the file control block.

Cannot read a record or an incomplete record is read from the input file:

- Empty file
- Invalid file handle (not likely if the file open function succeeds).

Cannot write to the output file:

- Disk full
- File has been captured by another process.
- Invalid file handle.

Cannot parse the input buffer from the file:

- Incomplete conversion of expected fields.

Common errors and the corresponding messages are listed in Table 11.

Table 11. Defined IO Errors

Error Name	Code	Description	Error Message
IO_SUCCESS	0	IO returned no errors	<none>
IO_NOFILE		File or directory not found	File does not exist
IO_BADHANDLE		Bad file handle	File corrupt
IO_FILE_EXISTS		Attempting to open file for writing that already exists	File already exists. Try again?
IO_NO_HANDLE		No more file handles	Cannot open file—too many files open
IO_DRIVE_FULL		No more disk space	Disk is full—cannot write to file
IO_CONVERT		Field conversion from text to numeric failed	Format error in line no. ___
IO_FILE_LOCK		File is locked by another user	File locked. Try again?

5.5.3 Uncertainty Processing Errors

Because the system design anticipates many occurrences of input data that cannot form a valid record for uncertainty processing, most exceptions in the uncertainty process will not halt the program execution. Rather, the uncertainty field will be marked as missing (-9900), and a descriptive code will be assigned to the record, which will be passed to the output file without interruption of the process.

An exception is if the SERI QC function returns an error code, which will halt processing. A SERI QC failure code indicates that an abnormal condition exists in the input data (corrupt date, time, etc.) or SERI QC has not been properly configured, which would likely affect much of the input data.

Table 12 describes all errors or abnormal conditions in the uncertainty process, including Function Uprocess() and Function Umerge.

Table 12. Defined Errors from Uncertainty Process

Error Code	Value	Description	Action
STATUS_VALID	0	Uncertainty successfully calculated	Assign value to uncertainty field; uncertainty code assigned STATUS_VALID.
ERROR_SERIQC	1	Processing halts	Error will be the function return value from Uprocess(), and it will prompt a popup message with the specific SERI QC error, input file line number, and a message that processing will halt.

Error Code	Value	Description	Action
ERROR_3_COMP	2	Less than three valid solar components (GHI, DNI, DHI) are available for processing	Mark uncertainty field as unassignable (-9900) and assign ERROR_3_COMP to code field.
ERROR_QCMAX	4	SERI QC assigned a flag above the specified SERI QC maximum	Mark uncertainty field as unassignable (-9900) and assign ERROR_QCMAX to code field.
ERROR_HI_Z	5	Zenith angle is above the specified maximum	Mark uncertainty field as unassignable (-9900) and assign ERROR_HI_Z to code field.
ERROR_LO_DNI	6	DNI (W/m ²) is less than the specified minimum	Mark uncertainty field as unassignable (-9900) and assign ERROR_LO_DNI to code field.
ERROR_ETR	7	ETR or ETRn equal zero [likely impossible for valid data but included for completeness]	Mark uncertainty field as unassignable (-9900) and assign ERROR_ETR to code field.
ERROR_KSPACE	8	Kn + kd = 0 [likely impossible for valid data but included for completeness]	Mark uncertainty field as unassignable (-9900) and assign ERROR_KSPACE to code field.
ERROR_HI_UNCERT	9	[To be defined later as maximum allowable valid uncertainty]	Mark uncertainty field as unassignable (-9900) and assign ERROR_HI_UNCERT to code field.

5.5.4 Popup Messages

Generic functions will be designed to handle various types of errors. In each instance, the function will receive a message string to be displayed prior to the user selecting a response.

Popup acknowledgement—This function will be called when an error occurs that requires an acknowledgement from the user before processing can continue. The popup will display a message with an “OK” button and wait for the user response.

```
Int Function Popup_OK (parameters: char message)
    Copy message to popup
    Display popup
    Return success
End
```

Popup yes/no—This function will be called when an error occurs that requires a yes or no decision. The popup will display the input message along with two buttons, labeled “yes” and “no,” and will wait for the user response. The function will return a code for yes or no.

```
Int Function Popup_YN (parameters: char message)
    Copy message to popup
    Display popup
    Return popup selection
End
```

5.6 File and Formats

The input and output file formats are rigidly defined to accommodate the expected resources available for programming. This approach requires users to provide input data in a specified, although common, format. The output format will also be in a specified format.

5.6.1 Input File Format

The input format is a CSV format that is compatible with the Microsoft Excel CSV (comma delimited) (*.csv), as listed in the Excel file export dialog window.

An optional single header line is allowed (user configurable), but the data lines must conform to the format and order described in Table 13. Each record must end with the LF or CR/LF characters, including the last record.

Table 13. Input File Format

Field Number	Description	Format Specifier
1	Date (month, day, year)	i/i/4i (single digits or leading zeros allowed for month and day, but <i>no spaces</i> ; year must be four digits)
2	Time (hour, minute) ^a	i:2i (24-hour time) (single digits or leading zeros allowed for the hour, but <i>no spaces</i> ; the minute should be two digits with leading zero. The time represents the ending time of the integrated data.)
3	GHI (W/m ²)	0.?f (floating point with optional decimal point and decimals)
4	DNI (W/m ²)	0.?f (floating point with optional decimal point and decimals)
5	DHI (W/m ²)	0.?f (floating point with optional decimal point and decimals)
6	Optional, ignored	Must be preceded by a non-numeric character

^a Time zone is not listed here; it is defined in the station's SERI QC .QC0 file specified by the user.

Example 1:

9/08/1989,14:06,894.2,704.2,134.5,Optional material

Example 2:

1/1/2020,9:19,376,0,360

5.6.2 Output File Format

The output file will consist of two sections: the *File Header* and the *Data Records*. All records will be in CSV format. The system will have a default *standard* format and an *extended* format, which includes additional parameters that can be optioned by the user.

The standard file header and output records are described in Table 14 and Table 15.

Table 14. Standard Output File Header

Field Number	Description	Literal Text
1	Date (year, month, day,)	"Date (YYYY-MM-DD)"
2	Time (hour, minute) ^a	"Time (HH:MM)"
3	GHI (W/m ²)	"GHI (W/m ²)"
4	GHI quality flag	"GHI SERI QC Flag"
5	GHI uncertainty value (±%)	"GHI Uncertainty (+/-%)"
6	GHI uncertainty code	"GHI Uncertainty Code"
7	DNI (W/m ²)	"DNI (W/m ²)"
8	DNI quality flag	"DNI SERI QC Flag"
9	DNI uncertainty value (±%)	"DNI Uncertainty (+/-%)"
10	DNI uncertainty code	"DNI Uncertainty Code"
11	DHI (W/m ²)	"DHI (W/m ²)"
12	DHI quality flag	"DHI SERI QC Flag"
13	DHI uncertainty value (±%)	"DHI Uncertainty (+/-)"
14	DHI uncertainty code	"DHI Uncertainty Code"

^a Time zone is not listed here; it will be as specified in the station's SERI QC .QC0 file.

Header line (wrapped here for presentation):

```
Date (YYYY-MM-DD),Time (HH:MM),GHI (W/m^2),GHI SERI QC Flag,GHI Uncertainty (+/-%),GHI Uncertainty Code, DNI (W/m^2),DNI SERI QC Flag,DNI Uncertainty (+/-%),DNI Uncertainty Code, DHI (W/m^2),DHI SERI QC Flag,DHI Uncertainty (+/-%),DHI Uncertainty Code
```

Table 15. Standard Output File Format

Field Number	Description	Format Specifier
1	Date (Year, month, day,)	4i-2i-2i (double digits with leading zeros for month and day; year must be four digits)
2	Time (hour, minute) ¹	2i:2i (24-hour time) (double digits with leading zeros)
3	GHI (W/m ²)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
4	GHI SERIQC quality flag	02i (two digits with leading zero)
5	GHI uncertainty value (±%)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
6	GHI uncertainty code	i (no leading zeros)
7	DNI (W/m ²)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
8	DNI quality flag	02i (two digits with leading zero)
9	DNI uncertainty value (±%)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
10	DNI uncertainty code	i (no leading zeros)
11	DHI (W/m ²)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
12	DHI quality flag	02i (two digits with leading zero)
13	DHI uncertainty value (±%)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
14	DHI uncertainty code	i (no leading zeros)

The extended format holds all the fields in the standard format but includes two more fields for system uncertainty and field uncertainty. The format is described in Table 16 and Table 17.

Table 16. Extended Output File Header

Field Number	Description	Literal Text
1	Date (year, month, day,)	“Date (YYYY-MM-DD)”
2	Time (hour, minute) ^a	“Time (HH:MM)”
3	GHI (W/m ²)	“GHI (W/m^2)”
4	GHI quality flag	“GHI SERI QC Flag”
5	GHI uncertainty value (±%)	“GHI Uncertainty (+/-)”
6	GHI uncertainty code	“GHI Uncertainty Code”
8	DNI (W/m ²)	“DNI (W/m^2)”
9	DNI quality flag	“DNI SERI QC Flag”
10	DNI uncertainty value (±%)	“DNI Uncertainty (+/-)”
11	DNI uncertainty code	“DNI Uncertainty Code”
12	DHI (W/m ²)	“DHI (W/m^2)”
13	DHI quality flag	“DHI SERI QC Flag”
14	DHI uncertainty value (±%)	“DHI Uncertainty (+/-)”
15	DHI uncertainty code	“DHI Uncertainty Code”
16	System uncertainty (±%)	“System Uncertainty (+/-)”
17	Field uncertainty (±%)	“Field Uncertainty (+/-)”

^a Time zone is not listed here; it will be as specified in the station’s SERI QC .QC0 file.

Extended header line (wrapped here for presentation):

```
Date (YYYY-MM-DD),Time (HH:MM),GHI (W/m^2),GHI SERI QC Flag,GHI Uncertainty (+/-%),GHI Uncertainty Code,DNI (W/m^2),DNI SERI QC Flag,DNI Uncertainty (+/-%),DNI Uncertainty Code,DHI (W/m^2),DHI SERI QC Flag,DHI Uncertainty (+/-%),DHI Uncertainty Code,Field Uncertainty (+/-%),System Uncertainty (+/-%)
```

Table 17. Extended Output File Format

Field Number	Description	Format Specifier
1	Date (year, month, day,	41-2i-2i (double digits with leading zeros for month and day; year must be four digits)
2	Time (hour, minute) ^a	02i:02i (24-hour time) (double digits with leading zeros)
3	GHI (W/m ²)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
4	GHI SERIQC quality flag	02i (two digits with leading zero)
5	GHI uncertainty value (±%)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
6	GHI uncertainty code	i (no leading zeros)
8	DNI (W/m ²)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
9	DNI SERIQC quality flag	02i (two digits with leading zero)
10	DNI uncertainty value (±%)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
11	DNI uncertainty code	i (no leading zeros)
12	DHI (W/m ²)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
13	DHI SERIQC quality flag	02i (two digits with leading zero)
14	DHI uncertainty value (±%)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
15	DHI uncertainty code	i (no leading zeros)
16	System uncertainty (±%)	0.1f (floating point with one decimal digit; leading zero for values less than 1)
17	Field uncertainty (±%)	0.1f (floating point with one decimal digit; leading zero for values less than 1)

^a Time zone is not listed here; it will be as specified in the station's SERI QC .QC0 file.

5.7 Report File Format

Following the processing of a data file, the system will present two statistical reports:

1. A popup window with a summary of the processing statistics
2. A text file with a complete summary of the processing statistics.

Both reports provide the user with summary statistics about the input file processed.

5.7.1 Popup Summary Statistics

The popup summary provides a glance at statistics to help the user judge whether the input file was processed correctly. The summary will appear as several lines in a popup text box on the

screen, and when the user has read it, it can be dismissed by clicking on a “done” button. Table 18 describes the format of the popup summary screen (titled “Summary Statistics”).

If the user has selected the extended file output, the averages for the U_{RADS} , system, and field uncertainties will be included in the report using lines 10–13 in Table 18.

Table 18. Popup Summary Statistics

Line Number	Description
1	Uncertainty processing report for <input data file name> <Station ID>
2	<blank>
3	Beginning: YYYY-MM-DD HH:MM, Ending: YYYY-MM-DD HH:MM Data records: [i]<record count>
4	Total eligible records: [i]<valid record count> (<[0.1f] percent>%)
5	Exceeded SERIQC max: [i]<failed record count> (<[0.1f] percent>%)
6	<blank>
7	GHI mean U95: [0.2f] +/- <GHI mean U95>% Standard deviation: [0.2f] <GHI Stedev>
8	DNI mean U95: [0.2f] +/- <DNI mean U95>% Standard deviation: [0.2f] <DNI Stedev>
9	DHI mean U95: [0.2f] +/- <DHI mean U95>% Standard deviation: [0.2f] <DHI Stedev>
10	<blank for extended processing>
11	U_{RADS} uncertainty mean [0.2f] +/-< U_{RADS} uncertainty mean>%
12	System uncertainty mean [0.2f] <system uncertainty mean>%
13	Field uncertainty mean [0.2f] <field uncertainty mean>%

Note: Field formats are in square brackets, e.g., [0.2f] indicates floating point with two digits past the decimal, and [i] indicates integer. Values are described in angle brackets.

Figure 7 shows a sample popup statistics summary report.

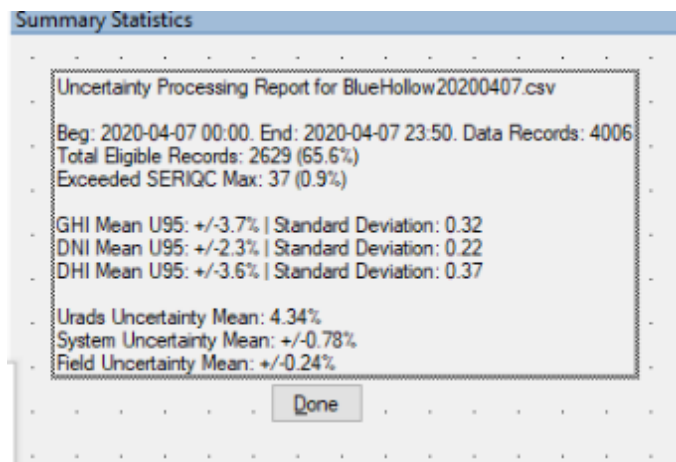


Figure 7. Sample popup summary statistics report

5.7.2 File Report Statistics

The system will produce a summary report text file after the uncertainty process is complete. This report will reside in the reports subdirectory and will be named according to the input file name by appending “_Report.txt” to the file name. For example, if the input file name is “BlueHollow20200407.csv”, the report file will be named, “BlueHollow20200407.csv_Report.txt”. If a file by that name already exists, prompt the user with “File <fname> exists. Overwrite?” The report format is described in Table 19.

If the user has selected the extended file output option, the information for the U_{RADS}, system, and field uncertainties will be included in the report using lines 24–27 in Table 19.

Table 19. Report File Format

Line Number	Description
1	Uncertainty processing report for <input data file name>
2	Processing date: <system date at runtime as MM/DD/YYYY HH:MM>
3	From YYYY-MM-DD HH:MM to YYYY-MM-DD HH:MM (%i-minute interval)
4	<blank>
5	System configuration:
6	GHI: s/n <instrument ID> RS: %0.2f uV/W/m ² U95+/-[0.2f]<U95> Cal date: <calibration date>
7	DNI: s/n <instrument ID> RS: %0.2f uV/W/m ² U95 +/-[0.2f]<U95> Cal date: <calibration date>
8	DHI: s/n <instrument ID> RS: %0.2f uV/W/m ² U95 +/-[0.2f]<U95> Cal date: <calibration date>
9	SERI QC max: <SERI QC Max>
10	Zenith angle max: [0.1f] <zenith angle max>
11	DNI min: [0.1f] <dni min>
12	<blank>
13	Input data records: [i]<record count>
14	Three-component records: [i]<valid record count> (<[0.1f] percent>%)
15	Above SERIQC max: [i]<failed record count> (<[0.1f] percent>%)
16	Above zenith angle Max: [i]<failed record count> (<[0.1f] percent>%)
17	Below DNI min: [i]<failed record count> (<[0.1f] percent>%)
18	Mathematically invalid: [i]<failed record count> (<[0.1f] percent>%)
19	Total eligible uncertainty records: [i]<valid record count> (<[0.1f] percent>%)
20	<blank>
21	GHI mean U95: [0.2f] +/- <GHI mean U95>% Standard deviation: [0.2f] <GHI Stedev>
22	DNI mean U95: [0.2f] +/- <DNI mean U95>% Standard deviation: [0.2f] <DNI Stedev>
23	DHI mean U95: [0.2f] +/- <DHI mean U95>% Standard deviation: [0.2f] <DHI Stedev>
24	<blank if extended processing specified>
25	U _{RADS} uncertainty mean [0.2f] +/-<U _{RADS} uncertainty mean>%
26	System uncertainty mean [0.2f] +/-<system uncertainty mean>%
27	Field uncertainty mean [0.2f] +/-<field uncertainty mean>%

Note: Field formats are in square brackets, e.g., [0.2f] indicates floating point with two digits past the decimal, and [i] indicates integer. Values are described in angle brackets.

The report file will be ASCII text to allow copying and pasting from the file into other applications, possibly even a formatted template report file. Following is a sample report file:

```
Uncertainty Processing Report for BlueHollow20200407.csv
```

Processing Date: 05/13/2020 13:24
From 2020-04-07 00:00 to 2020-04-10 23:59 (1-minute interval)

System Configuration:

GHI: s/n 200056 | RS: 9.04 uV/W/m² | U95 +/-3.43 | Cal Date: 20190505
DNI: s/n 180043 | RS: 10.14 uV/W/m² | U95 +/-2.05 | Cal Date: 20190505
GHI: s/n 180054 | RS: 9.56 uV/W/m² | U95 +/-3.66 | Cal Date: 20190505
SERI QC Max: 87
Zenith Angle Max: 80
DNI Min: 25

Input Data Records: 4006
Three-component Records: 3434 (85.7%)
Above SERIQC Max: 37 (0.9%)
Above Zenith Angle Max: 535 (13.4%)
Below DNI Min: 805 (20.1%)
Mathematically Invalid: 0 (0.0%)
Total Eligible Uncertainty Records: 2629 (65.6%)

GHI Mean U95: +/-3.7% | Standard Deviation: 0.32
DNI Mean U95: +/-2.3% | Standard Deviation: 0.22
DHI Mean U95: +/-3.9% | Standard Deviation: 0.37

Urads Uncertainty Mean: +/-4.7%
System Uncertainty Mean: +/-0.78%
Field Uncertainty Mean: +/-0.24%

5.8 Database Specifications

System databases will use a flat-file text format, which will eliminate the need for programmatically complicated database management software and interfaces. Nonetheless, care must be taken to protect the defined formats to prevent corruption or improper formats. The system will use three databases to facilitate (1) the station configuration and selection, (2) the instrument configuration, and (3) the integrated uncertainty processing system configuration.

5.8.1 Station Database

The station database will be implicit in the SERI QC station files that are created and maintained by the companion program QCFIT. To access the available stations, this system will look in the user-selected SERI QC directory for all files with the .QC0 extension. With a directory listing of all such files, the system will open each file and extract the Station ID and Station Name fields, which will be used to populate the pull-down list for the user station selection. When the user makes the selection, the Station ID will serve as the station identity. The Station ID will be passed to the SERI QC function call, which will use it to access its contents.

Because it is plausible that for larger network operations, a user might store the QC0 files for multiple stations in different directories, the SERI QC path will be one of the parameters entered by the user and saved as part of a system configuration.

5.8.2 Instrument Database

The instrument uncertainty values are created by the NREL SOLARUN application (Habte 2014), which provides an expanded uncertainty estimate based on the radiometer *calibration*, *model*, and *use*, where *use* is GHI, DNI, or DHI.

An instrument database will be created and maintained by the SOLARUN application as a collection of instrument information comprising the serial number, model, expanded uncertainty, calibration responsivity, calibration date, and calibration due date (Table 20). The entire instrument database will be a collection of *files*, with each file holding information for a single instrument.

Table 20. Instrument Table Record Description

Item	Description	Data Type
Instrument ID (serial number)	A unique ID for the instrument	Text (size?)
Instrument manufacturer	Manufacturer's trade name	Text
Instrument model	An instrument model from Table 21	Text
Instrument responsivity	Responsivity from latest calibration	Float
Calibration date	Date of the latest calibration	Text
Calibration due date	Date next calibration is due	Text
Measurement type	GHI, DNI, or DHI	Text
Instrument expanded uncertainty	Uncertainty as a balanced plus/minus, percentage	Float

The user must update the instrument table each time a new instrument model is added and each time an existing instrument is calibrated. As defined by the procedure, after the update, the instrument table record for an instrument will be exported to the instrument directory as a text file with a line for each field in the table. Following is a sample instrument database file (record for one instrument):

```
[sample file name: 190042_CHP1.txt]

ID: 190042
Model: CHP1
Mfgr: Kipp&Zonen
Rs: 9.42
Cal Date: 2023-06-06
Cal Due: 2024-06-06
Type: DNI
U95: 2.33
```

The system uses only the expanded (U_{95}) uncertainty for processing. The remaining instrument fields will provide instrument identification and calibration information for the reports. The instrument ID, model, and uncertainty fields are mandatory; all other fields can be listed as unknown if the information is unavailable.

5.8.3 System Configuration

A special directory will exist to hold the configuration files. These files retain the state of the system for future retrieval for similar processing. For example, a configuration file can be saved for each station for which future processing will be expected, allowing the user to select the file

name and put the system in the configuration for that station's identification, instrument collection, input data directory, and output file destination. This allows the user to more quickly set up and process many stations with different configurations.

The configuration files will follow a conventional .ini file format and will have a .ini extension. When retrieving a configuration, the system will look only at files in the configuration directory with the .ini extension.

When saving a configuration, the user will be prompted for a descriptive name to express the target configuration. This name can be any legal file name, considering length and allowed operating system characters, except for the two system configuration files noted here. For example, "Blue Hollow 2019.ini" could be used to describe the Blue Hollow station configured for 2019 data acquisition.

Two special system configuration files exist in the configuration directory:

- UncertDefaultsV001.ini (allows the user to reset the system to default settings)
- UncertPreviousConfig.ini (used automatically at system startup to reinstate the configuration when the system was last shut down).

A configuration file will have all the fields shown in Table 21.

Table 21. Configuration File Format and Contents

Field Number	Keyword	Description
1	station	The full name of the station
2	stationID	The abbreviated station identifier for SERI QC
3	inputFolder	The path, relative to the system folder, to the location of the input file
4	outputFolder	The path, relative to the system folder, to the location of the output file
5	SERIQCPath	The path, relative to the system folder, to the location of the SERI QC .QC0 files
6	GHIid	Serial number of the GHI instrument ^a
7	GHIrecord	The record number in the instrument database holding GHI uncertainty and calibration information
8	DNIid	Serial number of the DNI instrument ^a
9	DNIrecord	The record number in the instrument database holding DNI uncertainty and calibration information
10	DHIid	Serial number of the DHI instrument ^a
11	DHIrecord	The record number in the instrument database holding DHI uncertainty and calibration information
12	MaxQC	The maximum allowable SERI QC flag for uncertainty processing
13	MinDNI	The minimum threshold DNI for uncertainty processing
14	MaxZen	The maximum allowable zenith angle for uncertainty processing
15	extendedRpt	Flag to trigger an extended output report

^a For internal consistency, the ID will be checked against the ID in the record number.

The system will write all configuration files according to these conventions:

- An initial optional comment or comments starting with a semicolon
- A bracketed configuration section header
- All parameters listed as <parameter name> = <value> (string literals in quotes).

Following is a sample file:

```
; sample .ini file
[configuration]
station = "Blue Hollow"
stationID = "BH"
inputFolder = "\\network\2019Data\"
outputFolder = "\\network\2019Uncert\"
SERIQCPath = "network\data\SERIQC\"
GHIid = "14003"
```

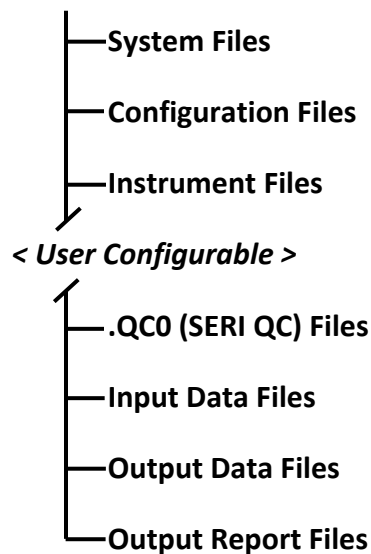
```
GHIrecord = 14
DNIid = "190042"
DNIrecord = 8
DHIid = "140033"
DHIrecord = 9
MaxQC = 87
MinDNI = 25
MaxZEN = 80
reportOut = 0
enhanceRpt = 0
```

When reading a configuration file, the system will expect the parameters in this order and will examine each line to confirm that the parameter name is correct and that a parameter of the correct data type follows the equals sign.

5.9 Directory Structure

The system requires that some data appear in specific non-configurable directories. Other information can be located elsewhere and can be user-configured.

Base Program Directory



6 Software Testing and Validation

When the system is nearing completion, a thorough software testing phase will begin with the intent to test all functional features. An outline of the testing procedure will be included here and will serve as a guide for specific tests to be developed during Deliverable 6.6.

It is likely that additional testing methods and parameters will be identified during the system development and testing phases. Such tests will be added to the test suite described in this section to form a final software testing document.

6.1 Module Functional Testing

For these tests, individual modules will be isolated in the program code and exercised with special driver code (where appropriate) with specific inputs that include and exceed the design parameters. These modules will include the user interface and the input file format errors.

6.1.1 *The User Interface*

- Each control will be exercised to determine if it returns the correct values in a click event.
- For controls without a click event function (such as checkboxes), testing will determine in either a debug print statement or in a stop-execution debugger to confirm that the state of the control is properly sensed.
- For pull-down lists that are generated from external sources (such as files in a directory), the list will be examined to determine if they contain all values from the source.
- The system will reject the command to process if all configuration parameters are not populated. As possible, each control in turn will be blanked or left unconfigured to ensure that the processing cannot continue.

6.1.2 *Input File Format Errors*

It is possible that the user could inadvertently select a file not related to the uncertainty processing. Primarily, the system should protect itself from input that causes an unstable or fatal condition. The SERI QC function will provide a first defense against out-of-bounds values, but some effort should be made to assess conformance to the specified format. This can take the form of ensuring that the date and time are in the correct format (e.g., '-' and ':' characters are in the correct position) and that there are the correct number of numeric values separated by commas in a record.

- The line should be parsed and the results should be checked for the correct number of expected values. Any condition that fails to assign the specified numeric value should halt processing with an alert to the user.
- A variety of input files can be created with simulated format error conditions. These can be configured to test each numeric field individually.
- Common file types—such as Word, Excel, or ZIP files—can be selected as part of the testing to ensure that such foreign formats are detected and rejected.

6.2 Functional Testing

6.2.1 *Boundary Value Testing*

The boundary testing uses input files with simulated data that exercise changes in program behavior at particular boundary conditions. These will include:

- Error values that trigger a change in SERI QC flagging that have predictable behavior in the uncertainty process
- Error values that trigger a predicted change in the resulting uncertainty
- An input file with synthetically derived three-component data with perfect coupling to test that the resulting uncertainty value for a measurement is neither more nor less than the radiometer uncertainty
- A synthetically generated three-component data set where perfect coupling is adjusted by a constant bias to ensure that the bias is represented as expected in the resulting measurement uncertainty.

7 Conclusions and Next Steps

A new SUNI algorithm consistent with GUM has been developed to assess the uncertainty of three-component solar irradiance measurements using SERI QC, an existing data quality assessment tool, and radiometer measurement uncertainties to determine the contribution of operational uncertainty to the overall uncertainty of solar resource data. The method has been successfully evaluated using 1-minute solar irradiance measurements collected during 2021 according to accepted best practices at SRRL in Golden, Colorado, and the NOAA network stations at Fort Peck and Penn State.

This document builds on the software requirements with software specifications that define the overall function of each section of the system. It provides details for the GUI, algorithm coding, database structure, error handling, and file formats.

This deliverable completes Task 4.2 in the project. With the knowledge that all sources of uncertainty have been identified, the programming work by NREL is the next step:

- Monitor and provide guidance for the Python code development.
- Modify specifications as the code development reveals problems or inconsistencies.
- Define the testing and validation requirements, and report on progress and required modifications to the existing algorithm.

References

- Coulson, Kinsell L. 1975. *Solar and Terrestrial Radiation, Methods, and Measurements*. Academic Press. ISBN 0-12-192950-7.
- Habte, A. 2014. Spreadsheet for estimating radiometer uncertainties available from the NREL Measurement & Instrumentation Data Center. https://midcdmz.nrel.gov/radiometer_uncert.xlsx.
- International Organization for Standardization (ISO). 2008. ISO/IEC Guide 98-3:2008(E): Uncertainty of Measurement – Part 3: Guide to the Expression of Uncertainty in Measurement (GUM: 1995). Geneva, Switzerland.
- Iqbal, M. 1983. *An Introduction to Solar Radiation*. Academic Press. ISBN 0-12-373752-4.
- Maxwell, E., S. Wilcox, and M. Rymes. 1993. *Users Manual for SERI QC Software—Assessing the Quality of Solar Radiation Data*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-463-5608. <http://www.nrel.gov/docs/legosti/old/5608.pdf>.
- Myers, D. R., T. Stoffel, A. Andreas, S. Wilcox, and I. Reda. 2000. *Improved Radiometric Calibrations and Measurements for Evaluating Photovoltaic Devices*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-520-28941. <https://www.nrel.gov/docs/fy01osti/28941.pdf>.
- Sengupta, Manajit, Aron Habte, Stefan Wilbert, Christian Gueymard, and Jan Remund. 2021. *Best Practices Handbook for the Collection and Use of Solar Resource Data for Solar Energy Applications: Third Edition*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-5D00-77635. <https://www.nrel.gov/docs/fy21osti/77635.pdf>.
- Shi, Y., and C. N. Long. 2003. “Preliminary Analysis of Surface Radiation Measurement Data Quality at the SGP Extended Facilities.” *Proceedings of the Thirteenth Atmospheric Radiation Measurement (ARM) Science Team Meeting*. Washington, D.C.: U.S. Department of Energy. ARM-CONF-2003. https://www.arm.gov/publications/proceedings/conf13/extended_abs/shi-y.pdf.
- Wilcox, Stephen, and Thomas Stoffel. 2022. *Evaluation and Suitability of Using SERI QC Software for Estimating Measurement Uncertainty*. Golden, CO: National Renewable Energy Laboratory. NREL/SR-5D00-82913. <https://www.nrel.gov/docs/fy22osti/82913.pdf>.
- Wilcox, Stephen, and Thomas Stoffel. 2023. *Developing the Proof of Concept for the SERI QC Flag Translation*. Golden, CO: National Renewable Energy Laboratory. NREL/SR-5D00-85589. <https://www.nrel.gov/docs/fy23osti/85589.pdf>.
- World Meteorological Organization (WMO). 1971. *WMO No. 8, TP 3: Guide to Meteorological Instruments and Observing Practices*. Geneva, Switzerland.

Appendix A. QC0 File

Site Identifier: SR2, Solar Radiation Research Laboratory

--- Latitude: 39.7400

-- Longitude: -105.1800

-- Time Zone: -7.0000

```

    ---Max Code---  ---Low Airmass---  --Medium Airmass--  ---High Airmass---
    KN  ----KT-----  Left  ----Right-----  Left  ----Right-----  Left  ----Right-----
           1  5 15 60  S  P  S  1  5 15 60  S  P  S  1  5 15 60  S  P  S  1  5 15 60

JAN: 87-99/00/00/00; 0-00 0-00/00/00/00; 4-08 2-17/00/00/00; 4-08 1-15/00/00/00
FEB: 87-99/00/00/00; 0-00 0-00/00/00/00; 5-08 1-17/00/00/00; 5-08 1-15/00/00/00
MAR: 85-99/00/00/00; 4-10 1-15/00/00/00; 5-08 3-17/00/00/00; 5-08 1-15/00/00/00
APR: 83-99/00/00/00; 4-10 1-18/00/00/00; 5-09 1-18/00/00/00; 5-08 1-15/00/00/00
MAY: 84-99/00/00/00; 4-10 1-17/00/00/00; 4-09 1-17/00/00/00; 4-08 1-16/00/00/00
JUN: 82-99/00/00/00; 4-10 1-16/00/00/00; 4-09 1-16/00/00/00; 4-08 1-14/00/00/00
JUL: 82-99/00/00/00; 4-10 1-16/00/00/00; 4-09 1-16/00/00/00; 4-08 1-13/00/00/00
AUG: 81-99/00/00/00; 4-10 1-16/00/00/00; 4-09 1-16/00/00/00; 4-08 1-14/00/00/00
SEP: 85-99/00/00/00; 4-10 1-15/00/00/00; 6-07 1-16/00/00/00; 5-07 1-12/00/00/00
OCT: 86-99/00/00/00; 0-00 0-00/00/00/00; 5-08 1-16/00/00/00; 5-06 1-14/00/00/00
NOV: 89-99/00/00/00; 0-00 0-00/00/00/00; 5-08 1-16/00/00/00; 5-07 1-13/00/00/00
DEC: 87-99/00/00/00; 0-00 0-00/00/00/00; 5-08 1-16/00/00/00; 5-07 1-14/00/00/00
```

DEFAULT CONFIGURATION

Integration (minutes): 1

Data Folder: d:\NSRDB2\Validation Data\SRRL

Plane: 1 Kt-Kd

3-Component Filter: -1.000000

Appendix B. SERI QC Flagging Convention

Table B-1. SERI QC Flagging Convention

From Maxwell et al. (1993)

Flag	Description										
00	Untested (raw data)										
01	Passed one-component test; data fall within max-min limits of K_t , K_n , or K_d										
02	Passed two-component test; data fall within 0.03 of the Gompertz boundaries										
03	Passed three-component test; data come within ± 0.03 of satisfying $K_t = K_n + K_d$										
04	Passed visual inspection; <i>not used</i> by SERI_QC1										
05	Failed visual inspection; <i>not used</i> by SERI_QC1										
06	Value estimated; passes all pertinent SERI_QC1 tests										
07	Failed one-component test; lower than allowed minimum										
08	Failed one-component test; higher than allowed maximum										
09	Passed three-component test but failed two-component test by >0.05										
10-93	<p>Failed two- or three-component tests in one of four ways.</p> <p>To determine the test failed and the manner of failure (high or low), examine the remainder of the calculation $(\text{flag} + 2)/4$.</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Rem</th> <th>Failure</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Parameter too low by three-component test ($K_t = K_n + K_d$)</td> </tr> <tr> <td>1</td> <td>Parameter too high by three-component test ($K_t = K_n + K_d$)</td> </tr> <tr> <td>2</td> <td>Parameter too low by two-component test (Gompertz boundary)</td> </tr> <tr> <td>3</td> <td>Parameter too high by two-component test (Gompertz boundary)</td> </tr> </tbody> </table> <p>The magnitude of the test failure (distance in K-units) is determined from: $d = (\text{INT}(\text{flag} + 2)/4)/100$.</p> <p>Examples and further discussion of the meaning of flags 10-93 are given in the text in this chapter and in Section 9.3, page 153.</p>	Rem	Failure	0	Parameter too low by three-component test ($K_t = K_n + K_d$)	1	Parameter too high by three-component test ($K_t = K_n + K_d$)	2	Parameter too low by two-component test (Gompertz boundary)	3	Parameter too high by two-component test (Gompertz boundary)
Rem	Failure										
0	Parameter too low by three-component test ($K_t = K_n + K_d$)										
1	Parameter too high by three-component test ($K_t = K_n + K_d$)										
2	Parameter too low by two-component test (Gompertz boundary)										
3	Parameter too high by two-component test (Gompertz boundary)										
94-97	Data fall into a physically impossible region where $K_n > K_t$ by K-space distances of 0.05 to 0.10 (94), 0.10 to 0.15 (95), 0.15 to 0.20 (96), and ≥ 0.20 (97).										
98	Not used										
99	Missing data										