For

*4027 Colonel Glenn Hwy, #210, Dayton, OH 45431, Ph 937.306.7193, Fax 937.306.7181*

*www.infoscitex.com*

**ARES OS**

**User Guide and Documentation**

Project Number: 60076729

Date: 24 August 2020

Requiring Office: THE OHIO STATE UNIVERSITY

174 Aerospace Research Center, 2300 W Case Rd. Columbus, Ohio 43235

**Revision History**

| **Revision** | **Doc Date** | **Description** | **Author/Submittal Date** |
| --- | --- | --- | --- |
| 001 | 11 August 2020 | Initial Release | Lauren Menke/  11 August 2020 |

Table of Contents

[1.0 Introduction 4](#_Toc48062087)

[2.0 Software Details 4](#_Toc48062088)

[3.0 Software Requirements 4](#_Toc48062089)

[4.0 Developer Setup 4](#_Toc48062090)

[5.0 User Guide 4](#_Toc48062091)

[5.1 User Interface (UI) 4](#_Toc48062092)

[5.2 Input 4](#_Toc48062093)

[5.3 Acceptable File Types/Formats 4](#_Toc48062094)

[5.4 System Dynamics 5](#_Toc48062095)

[5.5 Initial Uncertainty 6](#_Toc48062096)

[6.0 Distribution Types 6](#_Toc48062097)

[6.1 Gaussian 6](#_Toc48062098)

[6.2 Poisson 7](#_Toc48062099)

[6.3 Uniform 8](#_Toc48062100)

[6.4 Performance 9](#_Toc48062101)

[7.0 Notes 10](#_Toc48062102)

# Introduction

This program is a User Interface (UI) built and designed by IST. The UI takes input from the user via manual input or file upload and displays the output from the server it connects to. The UI will display errors and warnings if it detects something wrong with the input or connection to the server.

# Software Details

This UI program is a web based application, specifically an ASP .NET application running .NET Core 3.1. This application is Multi-Platform so it can run on just about any platform including but not limited to: Windows, Linux and MacOS.

# **Software Requirements**

In order to run the web server, .Net Core 3.1 will need to be installed on the computer.

# Developer Setup

When changing or editing the software, it is highly recommend to use the IDE “Visual Studio”, versions 2019 or later. This can be accomplished using the community edition. This is where the solution can be opened, built and ran. Note: It does utilize NuGet packages for third party needs.

# User Guide

## User Interface (UI)

[Placeholder for UI]

## Input

This section outlines the details regarding input types expected for this software. Most input values used can be expressions as well. This software accepts one file per type of input (dynamics, performance, etc), and one file for ALL of the inputs.

If using one file for all input, the software expects 3 sheets with the correct sheet name. All file types work for the per type basis. However, when using a single file for all inputs, that file must be an excel file. Each section has a specific expected sheet name. When the file is for a certain type of input, ignore the sheet name (if that file is an excel file).

**Note**: **Assume all input is case-insensitive unless stated in their section.**

## Acceptable File Types/Formats

* Excel Files(.xlsx, .xls, .xlsb)
* Comma Separated Values (.csv)
* Tab Separated Values (.tsv)
* Text Files (.txt). Note: contents must be either csv OR tsv but cannot be both, determination is based off the first occurrence of either a tab or a comma)

## System Dynamics

This part of the input is for constant variables (CC) and the system dynamic variables (SD). An SD variable has 2 or 3 fields. When this input is parsed, it will be assumed that anything with 2 fields will be an SD variable. Otherwise it will look for the type in the 3rd field. Hence, the line order for SD and CC variables does not make a difference and having the lines mixed is possible too.

**Expected sheet name**: “dynamics”

**Note**: ONLY the variable names which is acceptable in the first 2 fields are case sensitive.

Acceptable variable types:

* SD
* CC

Acceptable variable values:

* For example, a math expression that includes other declared variables or only a constant such as “4” or “54.67”; the parser can also recognize other variable names, “pi”, and math functions such as sin, cos, exp.

Acceptable variable names:

* Any name except math function names will work here. Math function names may cause issues when parsing.
* Variable names are **case sensitive**. When a variable name is mentioned in an expression, it is case sensitive.

Example input: per line -> {variable name}{variable value}{variable type}

SD variables:

x(1) x(2)\*sin(x(3))

rho rho0\*exp((h2 - x(1)\*R0)/h1) SD

CC variables:

mu 4.282837\*10^13 CC

rho0 0.0019 CC

## Initial Uncertainty

Initial Uncertainty contains a type and their parameters. Each type has more details and is further explained in section 6.0 (Distribution Types). A type declaration field and parameter declaration field are required for parsing. The line declaring the type must come before declaring the parameters.

Acceptable Type declarers:

* “type”

Acceptable Parameter declarers:

* "parameter"
* "par"
* "param"

**Expected sheet name**: “initial”

# Distribution Types

## Gaussian

This distribution type expects the mean parameter used to check the matrix size and expects the covariance parameter which is the expected diagonal matrix. The mean parameter is checked for expression validation. The parameter order does not make a difference, therefore the matrix or the mean can be presented first.

**Note**: This does not check the diagonal matrix for diagonality, but checks the matrix size.

Acceptable type names:

* "Gaussian"
* "gauss"
* "normal"

Acceptable mean declaring names:

* "mean"
* "micro"
* "mu"

Acceptable co-variance declaring names:

* "variance"
* "var"
* "covariance"
* "covar"
* "matrix"

Example Input:

{Type declaration}{Type name}

{Parameter declaration}{Mean parameter name}{Mean parameter 1}…{mean parameter x}

{Parameter declaration}{Covar parameter name}{Covar parameter 1}…{covar parameter x}

{Covar parameter 1}…{covar parameter x}

.

.

.

{Covar parameter x}

1.

Type Gauss

Parameter mean 80\*1000 3.5\*1000 (-1)\*2\*pi/180

Parameter covar 10\*1000000 0 0

0 0.01\*1000000 0

0 0 16/1000000

2.

“Type” normal

Parameter var 10 0

0 0.01

Parameter mu 80\*1000 3.5\*1000

## Poisson

This is another distribution type that can be handled. The lambda value can be a range value. For example, it may appear as 2:8 which means 2 through 8. Only the software can handle one lambda value.

Acceptable type names:

* "poisson"
* "pois"

Acceptable lambda declaring names:

* "lambda"
* "mean"
* "mu"
* "average"
* "avg"

Acceptable lambda values

* A value or expression with a result that must be at LEAST 0
* A range value

Example Input:

{Type declaration}{Type name}

{Parameter declaration}{Mean parameter name}{Mean parameter value}

1.

TYPE,pois,

Param,average,pi/180

2.

"TYPE" poisson

Parameter lambda 2:180

## Uniform

This is another distribution type that can be handled. The maximum cannot be smaller than the minimum or it will give an error. The minimum and maximum can be expressions as well. It does not matter which parameter line is first or last (minimum or maximum), as long as both exist.

Acceptable type names:

* "uniform"
* "uni"

Acceptable minimum declaring names:

* "min"
* "minimum"

Acceptable maximum declaring names:

* "max"
* "maximum"

###### Example Input

{Type declaration}{Type name}

{Parameter declaration}{Min parameter name}{Min parameter value}

{Parameter declaration}{Max parameter name}{Max parameter value}

1.

TYPE,uni,

Par,max,1000

Param,min,80

2.

type uniform

Parameter min pi

Parameter maximum 5\*50000

## Performance

Performance is where QOIs and their thresholds are input. For the thresholds, the maximum must be greater than the minimum, not equal to or less than the minimum. When thresholds are within 5% of each other, a warning will display. This warning can be accepted. The maximum and minimum can be expressions as well.

When naming QOIs, there are not many limitations on names chosen. The QOI value can be an expression or a constant. It can contain variable names that are parsed from the system dynamics section, meaning if the expression contains a variable, it is **case-sensitive.** There are no limits to the number of QOIs declared. The same are allowed as an acceptable variable expression that the system dynamics has for the QOI value.

**Note**: It will only have access to the LATEST variables parsed from system dynamics, if system dynamics was previously not ran when parsing performance, then it will just not know any variables if it detects them.

**Note**: The minimum must be set first before maximum or the numbers will be flipped (the maximum will be min and vice versa).

**Expected sheet name**: “performance”

Example Input:

{qoi name}{qoi value}{min threshold}{max threshold}

.

.

.

{qoi name}{qoi value}{min threshold}{max threshold}

Q1 0.25\*Cf\*rho\*vc^3\*sin(2)^3\*S 0.1 1

QName2 x(2)\*Vc/sqrt(gam\*Rstar^T) 0.001\*5 5.01

# Notes

This document is current as of 11 August 2020