Modernizing and validating WinDAM C: enhancing 64-bit software for earthen dam breach analysis

by

Prayas Gautam

B.E., Goa University, 2023

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Approved by:

Major Professor
Dr. Mitchell L. Neilsen

Copyright

© Prayas Gautam 2026.

Abstract

This study explores whether the modernized 64-bit application of WinDAM C developed by the U.S. Department of Agriculture–Agricultural Research Service (USDA-ARS) and Kansas State University, a program that routes a flood through a reservoir and estimates how overtopping and internal erosion of an earthen embankment dam progress toward a breach, is able to reproduce the results of the 32-bit legacy program. We enhanced the 64-bit application by adding several of the missing features of the legacy program as well as updating the data, screens, platform and installer of the stand-alone EFH-2 runoff program. The features extended included the automatic time-increment adjustment, hydrograph resampling, results and summary graphs, summary-table field selection, and Latin hypercube sampling of auxiliary-spillway soil properties. Seven example cases were run in the legacy WinDAM C 1.1.15 and in the 64-bit application (Version 1.1.17) using identical input files for each case with the automatic time-increment adjustment off and on. The results written to the output (`.OUT`) files by each simulator were compared. For the cases with the adjustment off, the peak outflow agreed within 0.1 percent for six of the cases and differed by 0.594 percent for the seventh case. For the cases with the adjustment on, the programs agreed as well as or better than without the adjustment for almost all quantities of interest in the two cases where both programs applied the adjustment. The difference of 0.594 percent for the seventh case decreased to 0.0316 percent. Where a difference could be traced, it came from the way in which the two interfaces prepared the input for the simulator or from the spillway simulator code; in the two cases where the simulators received identical input, the small differences came from the 32-bit and 64-bit builds of the dam simulator. The 64-bit application currently does not apply the adjustment to variable-time hydrographs. The agreement between the two versions of the software only indicates that the results are the same, not that they are correct.

Table of Contents

List of Figures vii

List of Tables viii

Acknowledgements ix

1 Introduction 1

1.1 Dams Built for Flood Control 1

1.2 Aging Dams and the Need for Breach Analysis 2

1.3 Research Question and Objectives 3

2 Breach Analysis and the Development of WinDAM C 5

2.1 Predicting Embankment Breach 5

2.2 Erosion Models in WinDAM C 7

2.3 Hydrology: EFH-2 and WinTR-20 7

2.4 WinDAM C and Its Legacy Application 8

2.5 The 2025 Modernization to 64 Bits 9

2.6 Why Two Builds Can Disagree, and Remaining Gaps 10

3 System Architecture and Implementation 12

3.1 Overview and Scope 12

3.2 State at the Start of This Work 13

3.3 Logical View: MVVM Structure 15

3.4 Data Model and Serialization 15

3.4.1 The .WDC simulator input format 16

3.4.2 JSON project representation . . . 17
3.4.3 Round-trip integrity . . . 18
3.5 Process View: Simulation Orchestration . . . 18
3.5.1 Invoking WinDAMSim and parsing .OUT results . . . 19
3.5.2 .D2D generation and SitesSim . . . 20
3.5.3 File and folder conventions . . . 21
3.5.4 Scenario: a complete simulation run . . . 23
3.6 EFH-2 Modernization . . . 24
3.7 Feature Extensions in WinDAM C . . . 25
3.7.1 Per-screen convergence settings . . . 25
3.7.2 Hydrograph resampling . . . 26
3.7.3 Results graphs . . . 27
3.7.4 Headcut advance visualization . . . 28
3.7.5 Summary table field selection . . . 28
3.7.6 Summary graphs across runs . . . 29
3.7.7 Latin hypercube sampling pipeline . . . 29
3.8 Development and Physical Views: Build and Deployment . . . 30
3.9 Summary . . . 31

4 Validation and Results . . . 32
4.1 Test Design . . . 32
4.1.1 Test Cases . . . 32
4.1.2 Programs . . . 33
4.1.3 Running the Cases . . . 33
4.1.4 Quantities Compared . . . 35
4.1.5 Agreement Criteria . . . 35
4.2 Input Files . . . 36
4.3 Results at the Same Time Increment . . . 36

4.3.1 Erosion Over Time 37
4.4 Automatic Time-Increment Adjustment 40
4.5 Sampling-Based Comparison 42
4.6 Differences Found 45
4.7 Summary 46

5 Conclusions and Future Work 47
5.1 Conclusions 47
5.2 Limitations 48
5.3 Future Work 49

Bibliography 50

A Validation Records 54
A.1 Programs Used in Chapter 4 54
A.2 File Comparison Command 54

B User Interface Screenshots 55

List of Figures

3.1 Components and the programs they run 13

3.2 Paths of project data 16

3.3 Start of a .WDC file 17

3.4 Steps of a simulation run 24

3.5 Steps of the sampling pipeline 30

4.1 Total outflow of 06-TH-OvertopBreach 39

4.2 Headcut station of 06-TH-OvertopBreach 39

4.3 Total outflow of 06-TH-OvertopBreach with the adjustment off and on . . . 42

B.1 PSH screen with the time-increment adjustment 55

B.2 SDH screen with the time-increment adjustment 56

B.3 FBH screen with the time-increment adjustment 56

B.4 Hydrograph screen with the time-increment conversion 57

B.5 View menu of the results graphs 57

B.6 Headcut Advance graph 58

B.7 Summary Fields dialog 58

B.8 Summary Graph Selections dialog 59

B.9 Summary graph across runs 59

B.10 Material Property Distributions dialog 60

List of Tables

3.1 Changes between the 2025 port and this work 14

3.2 Files of a project after a run 22

3.3 Build and installation of the two applications 31

4.1 Test cases 34

4.2 Programs used for the runs 34

4.3 Peak outflow at the same time increment 38

4.4 Reasons why output files differ 38

4.5 Simulator runs with the adjustment on 41

4.6 Legacy against 64-bit with the adjustment off and on 41

4.7 Files written in the sampling run 44

Acknowledgements

I would like to thank my major professor, Dr. Mitchell L. Neilsen, for giving me this opportunity and for his guidance, patience and support throughout this work. I also thank my committee members, Dr. Doina Caragea and Dr. Torben Amtoft, for their time and support.

I am grateful to Kansas State University, the Carl R. Ice College of Engineering and the U.S. Department of Agriculture for their support of this work, and to my lab mates for their help along the way.

Finally, I thank my parents for their love and encouragement.

Chapter 1

Introduction

1.1 Dams Built for Flood Control

The United States has been home to a large number of small earthen dams which have been built with assistance from the U.S. Department of Agriculture (USDA) for flood control.[1] Public Law 78-534, the Flood Control Act of 1944 started the small watershed program that produced them.[1] This was then followed by Public Law Protection and Flood Prevention Act of 1954 which expanded the program nationwide.[1] Construction averaged roughly around one dam per day for around two decades and over 11,000 small watershed dams were built over a period of seventy years with U.S. Federal involvement.[1] Hunt et al. report that nearly 90,000 dams are listed in the U.S. National Inventory of Dams (NID) and most of them are earth embankments.[2]. Many built since 1948 under the Small Watershed Program; The USDA Natural Resources Conservation Service (NRCS) is engineer of record for almost one-third of them.[2]

NRCS in Technical Release 210-60 (TR-60), *Earth Dams and Reservoirs* set the uniform minimum requirement for designing and planning earthen dams and their spillways.[3] Depending on damage, sudden breach or failure could cause therefore each dam is classed as having low, significant, or high hazard potential.[3] Loss of life could be caused by a failure of a high hazard dam and damage to farm buildings, farmland, and rural roads could be

caused by failure of low hazard dam.[3] An earthen dam usually has a principal spillway and an auxiliary spillway and they are sized by routing design hydrographs through them.[4]

The capacity of flood storage and principal spillway is determined by the principal spillway hydrograph (PSH).[4]. The principal spillway is sized to limit the frequency of flows in the auxiliary spillway.[4] The stability design hydrograph (SDH) is used to design larger flows passing through the auxiliary spillway.[4] The freeboard hydrograph (FBH) determines its minimum freeboard and integrity.[4]

1.2 Aging Dams and the Need for Breach Analysis

Overtopping and internal erosion are the leading cause of failure in earthen embankments and the aging of these dams has changed the risks they present[2] Overtime the situations and conditions that rise could cause both types of erosion.[2] The accumulation of sediment in the flood pool and changes in watershed conditions can leave a spillway with insufficient capacity.[1] The possibility of overtopping and, with it, the potential of loss of life and property downstream increases during extreme floods.[2] The possibility of failure by internal erosion increases due to long-term seepage, cracking, rodent activity and the growth of woody vegetation.[2] A further challenge that has arisen due to the aging of these dams are changes in hazard classification from low to significant to high.[2]

Existing dams that no longer meet safety criteria and are reaching the end of their design life are sponsored by NRCS Watershed Rehabilitation Program for rehabilitation.[5]

Dams have brought great benefits, but their failures have caused loss of life and property since ancient times.[6;7] Lower Van Norman Dam nearly failed in 1971 and Baldwin Hills Dam failed in 1964 .The state of California then asked dam owners to map the flood area due to dam failure which required steps to estimate outflow hydrograph of a breach on which little had been published.[8]. Failures of dams in mid-1970s led to reviews of dam safety program of Bureau of Reclamation, emergency planning with inundation maps were recommended by many of these reviews, and the Federal Guidelines for Dam Safety of 1979 also stated that such maps should be prepared.[8]

The rate of development, initiation time, depth, and the breach width must be accurately predicted where people live close to the dam and the details of the breach matter less where people at risk live far from the dam.[8] To classify the hazard potential of the dam, breach routings are used by NRCS to map the area that could flood due to dam failure.[3]. The parameter that most influences the loss of life is warning time as recognized by Wahl.[8]

The engineering problem has moved from design to analysis of existing structures due to aging of the dams.[1]. The USDA Water Resource Site Analysis Computer Program (SITES) which was developed mainly for design and could be applied to existing dams and spillways but it was developed when the computing capacity was far less.[1]. Neilsen, Temple and Hanson make this point as to why new software was needed. The estimation of how quickly an erosion will progress to a breach is required in order to analyse an existing dam.[8] The established tools are regression equation fitted to data from process-based models and historic failures built on the principles of sediment transport and support the floodplain mapping.[2]. However, a limited information about the physics of breach development, effects of cohesive soil properties and timing is offered by them.[2] Hut et al. concluded that in order to predict how homogeneous, cohesive embankments perform during internal erosion and overtopping new tools are needed.[2]

1.3 Research Question and Objectives

Addressing this need, the USDA Agricultural Research Service (ARS), NRCS, and Kansas State University developed the Windows Dam Analysis Module (WinDAM).[2] WinDAM officially meets the criteria by TR-60 as an acceptable process-based model for the peak discharge of a breach.[3] WinDAM C has been described as a useful tool for the prioritization and rehabilitation of existing dams by NRCS.[9] WinDAM C, its current version, its legacy 32-bit application, the 2025 modernization to 64-bits and the NRCS hydrology program EFH-2 is described in Chapter 2.

In this thesis we ask whether the modernization of WinDAM C reproduces the behavior of its legacy version. Three objectives follow.

1. Extend the C# and WinUI 3 version of the standalone EFH-2 application: its data, screens, platform, and installer (Section 3.6). Build its final version.

2. Add the missing capabilities to WinDAM C after the 2025 modernization work (Section 2.6): run-time adjustment of the time increment with per-screen convergence settings, hydrograph resampling, a headcut advance graph, and summary-field selection (Section 3.7).

3. Compare the legacy and modern builds case by case from identical input files, with and without time-increment control in the 64-bit build. Where feasible, compare results for sampled inputs (Chapter 4).

Chapter 2

Breach Analysis and the Development of WinDAM C

This chapter gives the background to the software described in Chapter 3. It reviews how embankment breach is predicted and how WinDAM C models it (Sections 2.1 and 2.2), and the hydrology behind EFH-2 (Section 2.3). It then describes the legacy WinDAM C application (Section 2.4) and its 2025 port to 64 bits (Section 2.5). Section 2.6 explains why two builds of the same simulator can disagree and lists the gaps that this thesis addresses.

2.1 Predicting Embankment Breach

The routing of the outflow of a failing dam is modelled in far more detail than the breach by a Dam-break flood model.[8] The user supplies the breach characteristics in many of them.[8] Wahl identifies three primary ways to predict the breach parameters: comparison with similar dams with historical failure, empirical regression equations fitted to case-study data, and physically based models of breach development, he also concludes these as insufficient.[8] The regression relations have great uncertainty, for large dams, case-study data are in short supply, and the existing physical models rest on a poor understanding of how breaches develop.[8] On a database of 108 failures, Wahl later quantified the uncertainty of regression

relations and notes that breach widths are uncertain by approximately one-third of an order of magnitude, failure times by up to an entire order of magnitude, and most peak-outflow predictions by one-half to one-full of an order.[10]

At that time, the best known physically based model was arguably the National Weather Service (NWS) BREACH model.[8] However, most of the breaching observed in the laboratory and field would not match the bed-load transport formulas used by such physically based models.[8] Wahl notes the main failure mechanism is a headcut formation which advances until the crest is breached with its initial formation at the toe of the downstream slope and no available breach model simulated it.[8] The headcut procedures developed by the U.S. Department of Agriculture (USDA) Agricultural Research Service (ARS) and Natural Resources Conservation Service (NRCS) for earth spillways were promising.[8]

These procedures for earth spillway came from laboratory and field work.[1] To improve earth spillway analysis, in 1983 the USDA Soil Conservation Service (SCS) and ARS formed a joint Emergency Spillway Flow Study Task Group (ESFSTG), its findings went into the DAMS2 program, and into the Water Resource Site Analysis Computer Program (SITES) in 1994.[1;11]

Worldwide, the studies of physical model of embankments produced breach models that more accurately reflect how cohesive embankments erode at small, intermediate and large scales.[2] The computational routines of overtopping and internal erosions were first developed in SIMBA (SIMplified Breach Analysis), a research tool created by the USDA ARS to analyze data and findings from large scale physical embankment failures and laboratory tests and then integrated into WinDAM.[2]. The recognized headcut migration, headcut width, and breach outflow for the primary ARS overtopping test were accurately simulated by WinDAM C.[2]. Atkinson and Neilsen later compared WinDAM C with DLBreach a physically based dam and levee breach model developed by Wu, on the internal-erosion failures of a 1.3 m laboratory dam and a 15.56 m dam that failed in the field.[12] The observed peak flow, time to peak, and breach width were more accurately produced by WinDAM C but the authors conclude that both the models can have a lower prediction accuracy.[12]

2.2 Erosion Models in WinDAM C

Embankment breach by internal erosion and overtopping are both modelled by WinDAM C and they assume a homogeneous earth embankment with non-erodible sides in a rectangular valley.[2] Soil erodes at the rate $k_d(\tau_e - \tau_c)$, where τ_e is the effective hydraulic stress. k_d is the detachment rate coefficient and τ_c is the critical shear stress which are entered by the users.[2]. The breach is followed in four stages by the overtopping model.[2]. Once any slope protection fails a headcut is formed on the downstream face. The headcut advances through the crest into the reservoir until the embankment is removed in the breach area and as the reservoir drains the breach keeps widening. The breach initiation time and the breach formation time are the end of stage two and stage three respectively.[2]

A horizontal, rectangular conduit through the embankment is placed and sized by the user where the internal erosion model starts. The widening of the conduit by the flow and the formation of the headcut at the downstream face where the headcut leaves are the two processes that happens at once and either of the process may dominate the breach.[2;13]. The models advance in time steps where the conditions at the start of each time step are held constant while the erosion during each time step is calculated.[2]

2.3 Hydrology: EFH-2 and WinTR-20

EFH-2 is an NRCS program that automates the procedures of Chapter 2 of the NRCS Engineering Field Handbook for small rural watersheds of 1 to 2,000 acres and newer versions utilize the hydrology procedures of the TR-20 computational engine and Part 630 of the National Engineering Handbook. EFH-2 is most frequently used by NRCS for small on-farm conservation practices.[14]. WinTR-20 is an event-based watershed-scale runoff and routing model.[15] Version 3.20 will also develop the principal spillway, stability design and freeboard hydrographs of Section 1.1 based on the criteria of NRCS Technical Release No. 60.[15]

The EFH-2 program uses the NRCS Runoff Curve Number (CN) Method to calculate the amount of runoff generated during a storm event.[14] Runoff is determined by both the

amount of rainfall received and the maximum amount of water that can be retained by the watershed; this is represented by a Curve Number ranging from 0 to 100.[14] The Curve Number is based upon the hydrologic soil group, land use, management, and hydrologic condition.[14] Peak discharge is also affected by the Time of Concentration (Tc), which is the time it takes for runoff to travel from the farthest point in the watershed to the outlet.[14] A shorter Tc results in a higher peak discharge.[14] Users of these types of programs provide information such as Drainage Area, Land Use, time of concentration, rainfall and can select a rainfall distribution and dimensionless unit hydrograph.[14]

EFH-2 appears in this thesis in two forms. One is the hydrology embedded in WinDAM C (Section 2.5). The other is the standalone application, whose soil table and rainfall distributions we changed (Section 3.6).

2.4 WinDAM C and Its Legacy Application

WinDAM is a series of modular software programs used to evaluate earthen embankment dams .[1;2] The software was created in increments; each increment added new modules based upon the latest research on embankment and spillway erosion.[1;2] WinDAM A routed a flood through the reservoir (including over the dam) and determined if the flood would cause failure of the grass or riprap on the downstream slope.[1;2] WinDAM A+, the first version to be widely used, included the auxiliary spillway erosion technology developed by SITES for up to three auxiliary spillways.[1] WinDAM B modeled the erosion of a homogeneous embankment from the time the slope protection failed until the embankment breached.[1;2;13] WinDAM C incorporated internal erosion and was the most recent version available in 2021.[2] NRCS released version 1.1.15 of WinDAM C in August 2024.[9] Hunt et al. state that both the modules and the program are referred to as models.[2] WinDAM C models the routing of an inflow hydrograph through a reservoir that has a principal spillway, up to three auxiliary spillways, and flow over the top of the dam.[2] In addition, WinDAM C determines when the slope protection on an overtopped embankment will fail and how rapidly erosion will proceed toward a breach.[2] WinDAM C performs the same type of analysis for erosion along a flow

path through an embankment.[2] WinDAM C only evaluates if a breach will begin for earthen auxiliary spillways.[2] WinDAM does not create the inflow hydrograph; the hydrograph is created by another program and brought into WinDAM.[2] Neilsen, Temple, and Hanson list SITES, HEC-HMS, HEC-RAS, and WinTR-20 as some of the programs used to create the hydrograph.[1]

Neilsen, Temple, and Hanson developed an interface to generate control files for another simulator and read back the files generated by the simulator.[1] They state that the well-defined interfaces allowed WinDAM to easily interface with other software.[1] The legacy WinDAM C works in this manner; the various WinDAM C programs exchange data via text files in the project directory.[16] The VB6 interface of WinDAM saves a project as a WinDAM Control file (.WDC), which is a text file containing keyword records.[16] The VB6 interface executes the dam simulator WinDamSim.exe on the .WDC file and, after the execution is complete, reads the results from the output file.[16] Each auxiliary spillway is saved as a .D2D file and is processed by the spillway simulator SitesSim.exe.[16]

The simulator usually employs the same time increment of the inflow hydrograph as its computational time step.[1;13] Neilsen, Temple, and Hanson have indicated that there is an option to reduce the increment by half until either the maximum stress or the reservoir water-surface elevation varies by less than one percent from one run to the next.[1] In the legacy source, this option is represented by a flag on the `HYD` record. If the flag is set, the application will rerun the simulator up to ten times, comparing peak outflow and peak water-surface elevation with limits specified on that record (Section 3.7.1).

2.5 The 2025 Modernization to 64 Bits

Microsoft stopped supporting the VB6 development environment on April 8, 2008.[17] The VB6 runtime is 32-bit only and will run on 64-bit Windows through the WOW emulation layer.[17] Microsoft will provide support for the VB6 runtime only for serious regressions and critical security issues; they recommend replacing all VB6 applications.[17] Comella-Dorda et al. defined modernization as a type of change that is more extensive than maintenance but

retains a large portion of the original system, and replacement as completely rebuilding the system.[18] Replacement requires extensive testing, since legacy systems have been thoroughly tested and optimized.[18] Scientific computing often uses the results of previously trusted programs as a basis for these tests.[19] Burugupalli modernized WinDAM C's VB6 interface to a 64-bit application developed in C# on WinUI 3 and the Windows App SDK utilizing the Model-View-ViewModel (MVVM) architecture.[16] The .WDC file served as an input for the simulators.[16] A project file in JavaScript Object Notation (JSON) format was created during the porting process and the application was packaged using the WiX Toolset.[16] EFH-2 procedures were incorporated into WinDAM C to allow for the generation of design hydrographs via WinTR-20.[16] This thesis continues this work. We contributed code of extensions to the same repository at approximately the same time; however, Section 3.2 separates the two projects.

Burugupalli used ten benchmark projects to test the modernization.[16] When ignoring whitespace, the simulator input files created with the new interface were identical to those created with the legacy application.[16] On the same inputs, the legacy 32-bit and new 64-bit simulators produced identical outputs in seven of ten cases; however, there were localized differences in three of ten cases.[16] She attributed these differences to the use of single-precision arithmetic versus double-precision arithmetic.[16] In two of the three cases where there were differences, reducing the time step resulted in closer agreement.[16]

2.6 Why Two Builds Can Disagree, and Remaining Gaps

The primary sources of error in a stable numerical solution, as listed by Oberkampf and Trucano, are: insufficient convergence in space, in time and of iterative procedures; computer round-off; and programming errors.[20] A typical method used to verify convergence is to refine the grid and the time step until there is little change in the important results.[20] Of the five sources listed above, two will be relevant when comparing two builds of WinDAM C: round-

off and time discretization. The time discretization is dependent upon the time step and how the application controls the time step (Section 2.4).

The differences between the two versions were reduced as well, according to Burugupalli, when the time step was decreased; she attributed this to the sensitivity of the time integration.[16] Her comparison did not include the legacy halving option since none of the ten benchmark projects used it. Although the 64-bit version of the application still included the option (Section 3.7.1), the simulator only ran once on each input file and thus the option had no effect. Therefore, based on current evidence, we cannot distinguish between numerical reasons for disagreement versus algorithmic ones.

The modernization of the software was missing several features (Section 3.2). For example, it could not convert a hydrograph with irregular time increments into constant time increment required by the halving option (Section 3.7.2); there was no summary graphs such as the headcut advance graph; and the command to select summary table fields did nothing. Additionally, this modernization was for WinDAM C only.[16] NRCS distributes EFH-2 as a standalone program with its own installation package.[14] The legacy standalone program is a VB6 application, version 2.0.13; another contributor started working on a C# and WinUI 3 version between December 2024 and April 2025 (Section 3.6). Chapter 3 describes how we addressed these gaps, and Chapter 4 tests whether the 64-bit application produces the same results as the legacy application.

Chapter 3

System Architecture and Implementation

3.1 Overview and Scope

The software referred to in this section is WinDAM C version 1.1.17, as it was in October 2026 after we made some modifications to the software that was ported in 2025. WinDAM C is a C# based WinUI 3 application. The other software referred to is the standalone EFH-2 application installer version 2.0.15.1. This is also a C# based WinUI 3 application. Both applications run as separate processes, passing an input file to each and then reading the files each writes (see Figure 3.1). WinDAM C runs the dam simulator, the auxiliary-spillway simulator, a sampling program, and WinTR-20. The EFH-2 application runs its own copy of WinTR-20.

We use Kruchten's 4+1 View Model to describe our architecture using four views (Logical, Process, Development, and Physical) and illustrate these views through scenarios.[21] The Logical View is described in Sections 3.3 and 3.4; the Process View is described in Section 3.5, with a scenario in Section 3.5.4; and the Development and Physical Views are described in Section 3.8. Section 3.2 compares the application before and after this work; Sections 3.6 and 3.7 describe the changes we made; Section 3.9 provides a summary of this chapter.

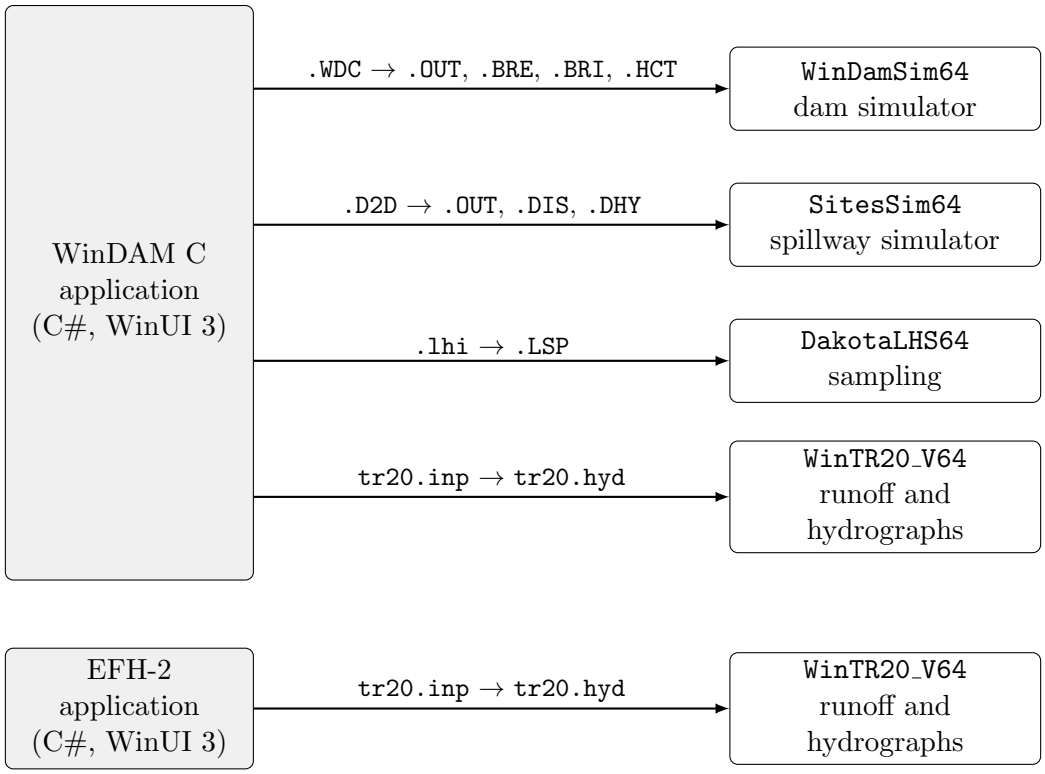


Figure 3.1: *The two applications and the programs they run, with the main input and output files of each. WinDAM C also runs* ***WinDamSim64*** *on the files derived from the* ***.WDC*** *file (Section 3.5.1).*

3.2 State at the Start of This Work

Table 3.1 is a comparison of the application at the time Burugupalli left it (version 1.1.16, October 2025) and the application as described in this chapter as of October 2026. The application described in this chapter is version 1.1.17. All changes to the application's source code were made during this time frame; however, changes to the simulator executables and test data during this time frame were made by another contributor and are therefore not included in our work. The application's user interface libraries and installer definition were not changed.

We have also worked on three features that were available in the 2025 version: importing hydrographs from WinTR-20, HEC-RAS, and HEC-HMS. We added these features between March and April 2025. Burugupalli listed importing from TR-20 and HEC-RAS as possible future developments.[16] Sections 3.7.1 to 3.7.7 will provide detailed descriptions of each of

Table 3.1: *Components of WinDAM C changed between the end of the 2025 port (version 1.1.16, October 2025) and the version described in this chapter (October 2026). We made all the changes listed.*

Component	Version 1.1.16 (2025)	Version 1.1.17 (2026)
Target framework	.NET 7, out of support since 14 May 2024[22]	.NET 10, a long-term support release supported to 14 November 2028[22]
Time-increment adjustment during a run	Setting saved for the main hydrograph, but the simulator ran once per input file	Legacy halving loop ported; up to ten simulator runs per input file
Convergence settings on the PSH, SDH, and FBH screens	None	One setting per screen, saved to and read from the project files
Variable- to constant-time conversion	Not implemented; the button only changed the view	Linear resampling onto a constant time increment
Results graphs	Graphs of input data; hydrograph and spillway-erosion graphs from the summary table; *Graphs* menu entry inactive	19 graph types built from the run's input and output files, including headcut advance
Summary-table field selection	Menu entry inactive	Field-selection dialog
Summary graphs across runs	Button inactive	Port of the legacy summary graphs
Latin hypercube sampling (LHS) of spillway properties	Existing sample files read by the summary table	Samples generated with Dakota; one `.D2D` file per sample; `SitesSim` runs in parallel
SITES hydrograph import	`.SHY` files only	`.DHY` reader ported from the legacy code

the major extensions.

3.3 Logical View: MVVM Structure

The application will follow, at a high level, the MVVM architecture described by Burugupalli.[16] There are fifteen screens that get all of their data from one application-wide view model called `MainViewModel`. This view model uses the standard `INotifyPropertyChanged` interface to notify the user interface when there are any changes. All of the project data is stored in a separate class called `MainDataModel` and is serialized into JSON format. Each EFH-2 screen has its own view model created using the CommunityToolkit MVVM library. Most of the XAML views use run-time `Binding` expressions (569) instead of compiled `x:Bind` expressions (66) to bind to their respective view models.

The code deviates from this separation in one significant way: the simulation is controlled by the main window's code-behind instead of being controlled by a view model. The method used to run a project (`RunSimulationMethod`) is located within `MainWindow.xaml.cs`, which contains 3,666 lines of code is used to start the engines and sampling pipeline. Additionally, the code-behind for the summary table is 2,702 lines long. Many of the new classes we created reside outside of these two files. Classes such as the time increment loop, convergence setting parser, hydrograph resampler, SITES hydrograph reader, summary graph builder, and sampling classes do not reference the UI framework or `MainViewModel`. Instead, they are invoked from the code-behind of the screens that utilize them or, in the case of the convergence setting parser, from the code that reads and writes project files. The only class that does reference the UI framework and `MainViewModel` is `GraphManager`, which generates the results graphs and manages some aspects of the summary table page.

3.4 Data Model and Serialization

A WinDAM C project exists in two forms: a JSON file, which the application uses to store and reload the project, and a `.WDC` file, which the simulators read. Figure 3.2 shows how data move between them.

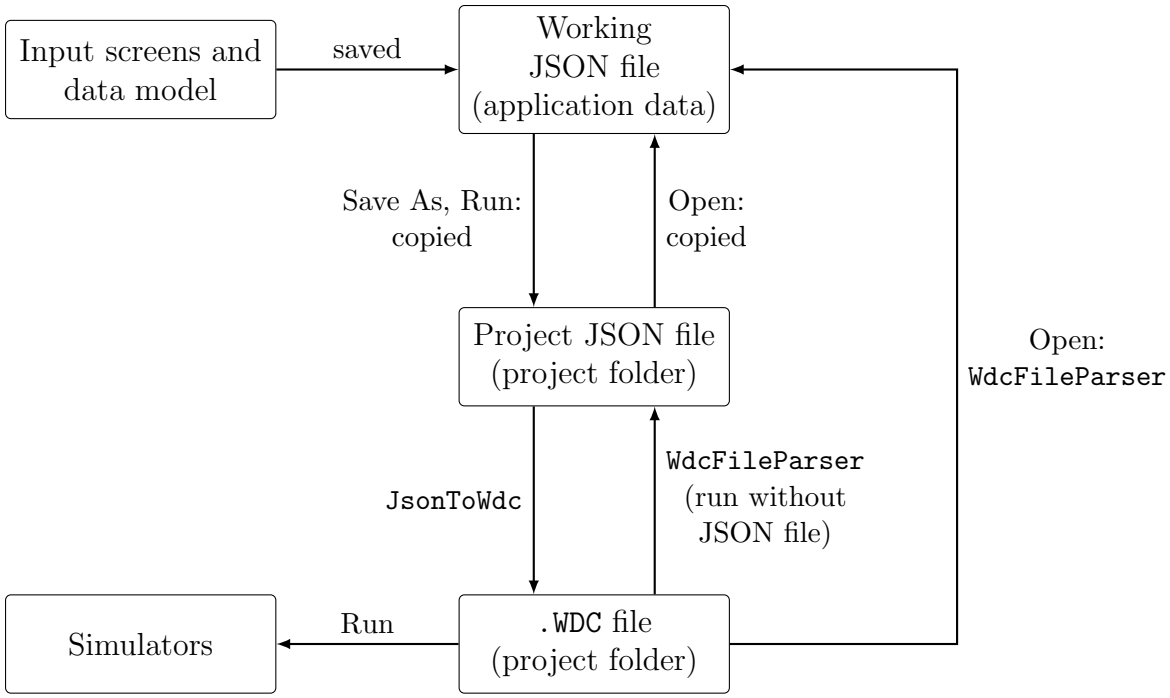


Figure 3.2: *Paths of project data between the input screens, the JSON files, and the* `.WDC` *file. Arrow labels name the action or class that moves the data; Section 3.5 describes the run.*

3.4.1 The .WDC simulator input format

The `.WDC` file is a line-oriented text file composed of records. Each line begins with a record keyword in the first ten columns. The remaining columns on the line contain the values associated with the record. These values are stored in fixed-width fields of ten columns. A table, such as the inflow hydrograph, will continue onto subsequent lines until an `ENDTABLE` line is encountered. Figure 3.3 illustrates the beginning of one of the benchmark projects: the `WINDAM` header with a date and a title; the `OPTION` record which selects the analysis; comment lines marked with an asterisk; and the `HYD` record followed by the first time–discharge pairs of the hydrograph.

The `HYD` record contains the settings for the hydrograph in fixed columns: the time increment is located in column 21; a flag indicating whether the time increment should be automatically adjusted is located in column 41; and the type of table is located in column 42 (`C` for a constant and `V` for a variable time increment). If the flag is set, three convergence limits follow in columns 51 to 80 (Section 3.7.1). A project whose hydrographs are generated

```
12345678901234567890123456789012345678901234567890123456789012345678
WINDAM    03/07/2024Example 02
OPTION    SIMPLE    EARTH               NOBREACH
*         WinDAM B Example 02 - Existing Dam  06/22/2012
...
*         END of input comments------------------
HYD       SDH Stab  0.063               0V
          0         1.35
          0.063     4.54
...
```

Figure 3.3: *Start of the* `.WDC` *file of benchmark project 02-SDHstability. The first line is a column ruler added for reference; "..." marks omitted lines.*

from EFH-2 stores each design hydrograph in its own record (`HYD_PSH`, `HYD_SDH`, or `HYD_FBH`) and stores a record containing the EFH-2 inputs for that hydrograph adjacent to it. We have included the convergence settings in these three records in the same columns as in the `HYD` record and have also included records that store the sampling configuration of each auxiliary spillway (Section 3.7.7).

3.4.2 JSON project representation

The JSON file used in the 2025 modernization contains the project in the same format as the data model of the application (`MainDataModel`), and is created using the Newtonsoft.Json library.[16] When a project is open, all screens write their data to a working copy of this file located in the user's application-data folder; there will be one working copy per instance of the application that is currently running. When you save or run your project, the working copy is copied to your project folder (with the same name as your project) and then the `.WDC` file is generated from it. Since we have added fields to the data model (convergence settings for each hydrograph screen and sampling configuration for each auxiliary spillway), these fields are contained within both files.

3.4.3 Round-trip integrity

Two converters are used to convert between the two formats; `JsonToWdc` converts the `.WDC` file from the JSON file, while `WdcFileParser` reads a `.WDC` file back into the data model. A project may be opened using either format; opening the JSON file will copy it into the working file, and opening a `.WDC` file will execute the parser. Additionally, we have configured the parser to run when simulating a project which has a `.WDC` file but does not have a JSON file, so that a JSON file is generated for it.

Burugupalli compared the `.WDC` files generated by the new interface with those generated by the legacy application for ten benchmark projects and determined that they were identical once differences in whitespace were ignored.[16] This comparison evaluated only the writer and none of the ten projects contained EFH-2 hydrographs or sampling configurations, therefore the records containing these were not evaluated. There is currently no automated test in the repository to evaluate whether reading a `.WDC` file and then writing it again results in an identical file. Chapter 4 evaluates the written files again.

3.5 Process View: Simulation Orchestration

The process view of Kruchten's 4+1 Model views a system as a group of executing programs, and focuses on how the system manages concurrency, fault tolerance, and the overall integrity of the system.[21] In WinDAM C, the programs are the application and the two simulators, and the programs communicate via files. The application provides each simulator with the location of an input file and then reads the files that the simulator produces. A single execution of a run consists of four steps. The application creates the input files for the simulators, executes the dam simulator on each input file, summarizes each output listing, and then executes the auxiliary-spillway simulator on each spillway and hydrograph; if sampling has been configured, the sampling pipeline will be executed next (Section 3.7.7). The application executes the simulator processes sequentially and waits for each process to complete before executing the next process; however, in the sampling pipeline, up to four simulator processes

may be executed simultaneously. Each of the following sections describes how these steps were implemented in the version of October 2026, and Section 3.5.4 illustrates one complete execution of a run.

3.5.1 Invoking WinDAMSim and parsing .OUT results

The project may be run using up to four inflow hydrographs which will be selected on the project information screen. Of these, three are the Principal Spillway, Stability Design, and Freeboard Hydrographs (PSH, SDH, and FBH) as defined by NRCS criteria for earth dams.[15] The fourth hydrograph, labeled "Others", is the hydrograph contained within the main `HYD` record of the project. The selection of hydrographs is represented by flags on the `OPTION` record. Since the dam simulator can only read one hydrograph from each input file, prior to each simulation, the application creates an input file for each selected design hydrograph from the `.WDC` file. Each created file contains all records preceding the main hydrograph; the design hydrograph's record (`HYD_PSH`, `HYD_SDH`, or `HYD_FBH`; see Section 3.4.1) renamed to `HYD`; and all records following the main hydrograph up to the EFH-2 records. The `.WDC` file is simulated when "Others" is selected or when no hydrograph is selected. Additionally, during this process, each auxiliary-spillway block in the `.WDC` file is copied into a separate file (see Section 3.5.2). Both processes were implemented in the 2025 modernization.

The input file will be processed by `ConvergenceRunner`, a new class we created. This class will start `WinDamSim64.exe` as a separate process from the `Assets` folder of the application. It will run without a window. The only argument for `WinDamSim64.exe` is the path to the input file, and the folder where the input file resides is the working directory. `WinDamSim64.exe` will write its results next to the input file with the same name. These results include: the `.OUT` listing and files containing details of breach (`.BRE`), internal erosion (`.BRI`), and breach progression (`.HCT`) depending on the type of analysis. If the `HYD` record does not require an adjustment of the time increment, then `ConvergenceRunner` will start `WinDamSim64.exe` once. However, if the `HYD` record requires an adjustment of the time increment, then `ConvergenceRunner` will use the halving loop described in Section 3.7.1. In either

case `ConvergenceRunner` takes the console output from `WinDamSim64.exe`, discards the console output as long as `WinDamSim64.exe` is running, and then waits for `WinDamSim64.exe` to finish. The 2025 version took the same output but did not read it. According to Microsoft's documentation, a parent process that waits for a child process to complete while taking its output can be deadlocked if the child process fills the stream.[23] Since `ConvergenceRunner` has no time limit and does not check the exit code, there are only files created by `WinDamSim64.exe` to determine if a run was successful.

The application takes the `.OUT` file which is produced after every simulator run and converts it into a `.DIS` summary. This summary is then used to create the summary table, as it was done in the 2025 modernization.[16] The conversion process is performed by the `ParserOutFileToDisFile` class. This class creates a fixed list of 49 lines, each containing one value, by searching through the listing for labeled values (e.g., peak inflow, maximum pool elevation, peak outflow, maximum flow through each spillway) and counting the number of warnings generated by the simulator. If a value is not located, it is written as `N/A`. In the 2025 version, the lines allocated for internal erosion were always `N/A`; however, we have now populated these lines based on the legacy parser. They currently contain the peak flow through the internal-erosion conduit; the width, height, invert elevation, station, and length of the conduit; the time when flow begins through the conduit; the parameters of the internal-erosion model; and the time when the conduit collapses, which is obtained from the `.BRI` file. Any errors occurring during the conversion process are caught and will not be displayed in the user interface. So, if there is no `.OUT` file available, the resulting `.DIS` file will either be empty or will have the results from a previous run.

3.5.2 .D2D generation and SitesSim

The second simulator, SitesSim, will analyze the auxiliary spillways after the dam simulator runs. Like in the 2025 modernization, the application will create one input file in the `.D2D` format for each auxiliary spillway. Each `.D2D` file will contain the spillway's block from the `.WDC` file and the discharge through that spillway calculated by the dam simulator. These

files will be placed in a subfolder named after the project file.[16] A spillway will have a block in the `.WDC` file only if a stability or integrity analysis is chosen for that spillway. The subfolder will be deleted and recreated each time the application runs. The application will then start `SitesSim64.exe` on each `.D2D` file in turn, as a separate process without a window and with the path of the file as the only argument, and wait for each run to complete. SitesSim will write its listing (`.OUT`) next to each input file. Since every spillway block includes a `GRAPHICS` record, SitesSim will also write a summary file (`.DIS`), which the summary table reads directly, and a hydrograph file (`.DHY`).

3.5.3 File and folder conventions

The files for each project are located in a single project folder. When you create a new project folder, it will contain two files: `SUMMARY.IDE`, which contains all of the runs that appear in the summary table; and a hidden, read-only file called `owner`. The presence of the `owner` file indicates that the folder is a WinDAM project. If a folder does not have the `owner` file, then WinDAM will not allow you to open that folder as a project nor will it allow you to create a project within an existing folder. You can save your project file only in the project folder, and you cannot use any of the following suffixes for your project file name: `_PSH`, `_SDH`, or `_FBH` (the suffixes used for the derived input files). Table 3.2 shows the files that are created by a run in both the project folder and the subfolder for auxiliary spillway analysis. We have maintained these conventions since the 2025 version; however, we have added files for the halving loop and the sampling pipeline (Section 3.7.7).

At each simulation run, the program will add the names of all input files (simulated) to `SUMMARY.IDE`, if they have not been previously added to the list. It will remove from the list the names of hydrographs which are no longer selected. If a design hydrograph is no longer selected, it will also delete the files associated with this design hydrograph. The summary table contains one column for each name in the list, and will fill each column from the `.DIS` file of that name. Its auxiliary spillway table will read the `.DIS` files created by SitesSim in the folder shared by the project file and its design hydrograph files.

Table 3.2: *Files of a project after a run. N is the name of the project file; F is the name of a simulated input file (N, N_PSH, N_SDH, or N_FBH); i is the number of an auxiliary spillway and n the number of hydrograph points of a pass. Files of the sampling pipeline are described in Section 3.7.7.*

File	Folder	Written by	Content
`SUMMARY.IDE`	Project	Application, at creation; updated at each run	Runs shown in the summary table
`owner`	Project	Application, at creation	Marks the folder as a project folder
N`.json`, *N*`.WDC`	Project	Application, on Save As and Run	Project data and simulator input (Section 3.4)
N`_PSH.PSH`, *N*`_SDH.SDH`, *N*`_FBH.FBH`	Project	Application, before each run	Input for one design hydrograph
N`.AS1` to *N*`.AS3`	Project	Application, before each run	One auxiliary-spillway block each
F`.OUT`, *F*`.BRE`, *F*`.BRI`, *F*`.HCT`	Project	`WinDamSim64`	Listing; breach, internal-erosion, and breach-progression details
F`.DIS`	Project	Application, after each simulator run	49-line summary of *F*`.OUT`
Fn`.D2D` and its outputs	Project	Application and `WinDamSim64`	One pass of the halving loop; kept after the run
Subfolder *N*	Project	Application, at each run	Auxiliary-spillway analyses; deleted and created again at each run
Fi`_0.D2D`	*N*	Application	Input of `SitesSim64` for spillway *i*
Fi`_0.OUT`, `.DIS`, `.DHY`	*N*	`SitesSim64`	Spillway listing, summary, and hydrograph

3.5.4 Scenario: a complete simulation run

The diagram shown in Figure 3.4 illustrates the process of running a simulation when you select the "Run Simulation" option. When you choose to run a simulation, if the project has not been saved, the program will ask you to name the file in your project directory. If the project has been saved, the program will ask you if you want to save your current work prior to running the simulation. If you choose to save your work, the program will copy the JSON file you are working with to the JSON file for your project and create a `.WDC` file based upon that JSON file (see Section 3.4). If you choose not to save your work, the program will use the `.WDC` file that is currently on your computer. After saving your work (or choosing not to), the program will generate the input files for the design hydrographs and the auxiliary spillway files (see Section 3.5.1), update `SUMMARY.IDE` (see Section 3.5.3), and run the dam simulator using `ConvergenceRunner` for each input file you have selected. Each input file will be converted into a `.DIS` file. Any messages generated by the halving loop (such as "did not converge" or "hydrograph has a variable time increment") will appear in a single dialog box. Once all of the `.DIS` files have been created, the program will create the `.D2D` files and run SitesSim on each of those files (see Section 3.5.2). If the project's JSON file does not exist, the program will create it. Finally, if a sampling pipeline is configured, the program will execute it. At the conclusion of the execution of the sampling pipeline, a dialog box titled "Build Successful" will appear. This dialog box will appear regardless of whether or not the simulators produced valid results. The "Simulate All Files" command performs the same sequence of actions as described above; however, instead of performing them only once, the program will perform them sequentially on each `.WDC` file located within the project's directory. Each `.WDC` file will be simulated as it exists on disk and will not be re-created from its corresponding JSON file.

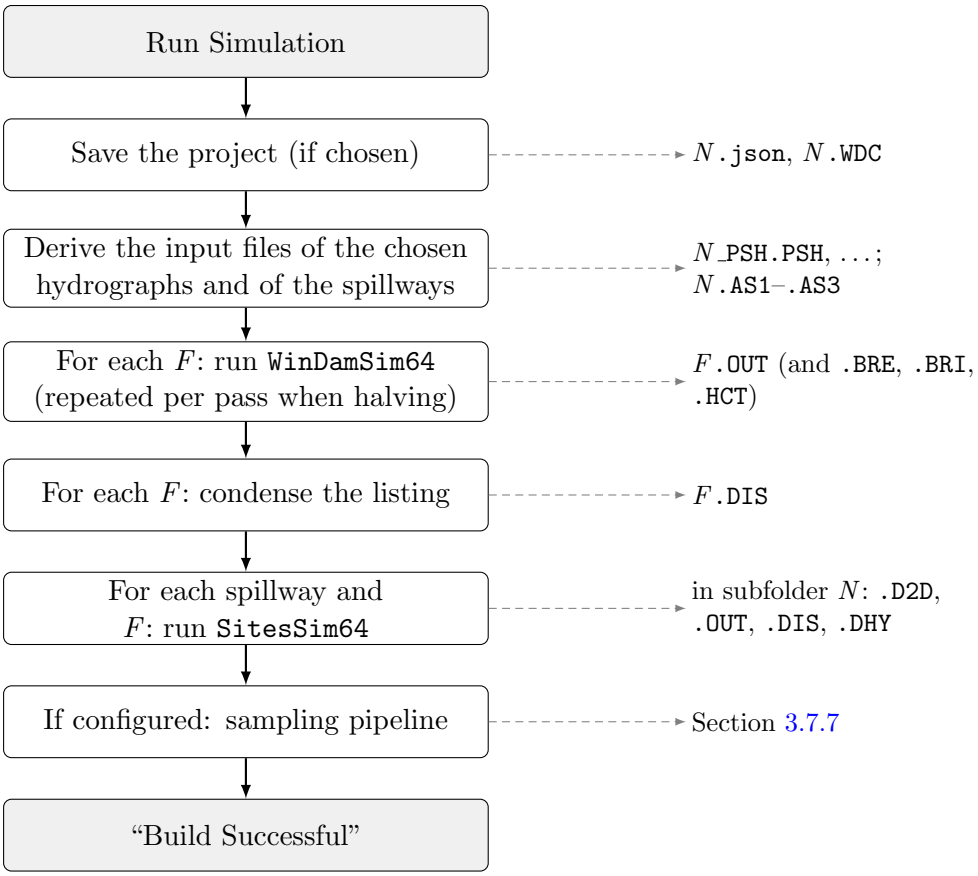


Figure 3.4: *Steps of one Run Simulation in the version of October 2026, with the files each step writes. N and F are as in Table 3.2. The project is saved only when the user chooses to. When the HYD record of F requests adjustment of the time increment, ConvergenceRunner runs WinDamSim64 once for each pass (Section 3.7.1). The update of SUMMARY.IDE, the dialog of halving messages, and the creation of a missing JSON file are omitted.*

3.6 EFH-2 Modernization

The stand-alone EFH-2 program uses the NRCS method described in Section 2.3 to calculate runoff and peak flow. The original version of the program was developed in VB6 and is available as version 2.0.13 in the repository. Between December 2024 and April 2025, another developer started developing a C# and WinUI 3 version of the program. This section describes only the modifications we made to that version between September 2025 and October 2026; the resulting installer is version 2.0.15.1. WinDAM C includes EFH-2 screens created by Burugupalli.[16] These screens are part of the WinDAM C project and do not reference the code of the stand-alone EFH-2 program.

We modified two aspects of the data the program reads. We replaced the soil table used to determine the hydrologic soil group of a soil (which contained 880 records) with a table containing 70,242 records provided by another contributor, and reorganized the lookup page. We also added five rainfall distribution tables (`INW_A` through `INW_E`) and modified how the program finds the file associated with a given rainfall distribution type. Previously, the program would build the file name from the name of the type; however, this approach would fail if the two were different.

We also changed the basic-data, rainfall and discharge, hydrograph, soil-group, and print screens, and updated the help pages and the user manual supplied with the application. We migrated the application from .NET 6 to .NET 10 (the version WinDAM C uses as described in Section 3.2) and from version 1.6 to version 1.7 of the Windows App SDK. The project file referenced a publish profile for 64-bit builds that was missing; we created it so that a self-contained build will be published into the directory where the installer is created. We updated the installer's version to 2.0.15.1 and enabled upgrades between packages of the same version.

3.7 Feature Extensions in WinDAM C

3.7.1 Per-screen convergence settings

The 2025 version of the main hydrograph screen allowed users to change the computational time increment while running a simulation; however, because the simulator only ran once per input file, this feature did not work. The three EFH-2 hydrograph screens did not allow users to do this.

We have now included this feature along with three limitations on all three screens (PSH, SDH, FBH): the maximum allowable percent change in peak discharge, the maximum allowable change in water surface elevation, and for drawdown runs, the maximum allowable difference in water surface elevation at the reference time (Figures B.1, B.2, and B.3). Default values are 10 percent, 0.02 feet, and 0.02 feet respectively. All settings from each screen will

be saved in both the JSON file and in columns 41-43 of the `HYD_PSH`, `HYD_SDH`, or `HYD_FBH` records just like they were in the `HYD` record (see Section 3.4.1). When the parser reads the input file created for a design hydrograph, those columns remain intact (see Section 3.5.1), thus allowing the setting to be passed through to the simulation.

The `ConvergenceRunner`, a new class is used to make these adjustments. We have taken and modernized this from the legacy applications run loop. If an input file requests for it, and your hydrograph has a constant time increment, then the convergence runner will run the simulator up to ten times. On the first run, the hydrograph will be resampled at the same time increment as the hydrograph. Each subsequent run will halve the time increment, thus doubling the number of intervals, and create a temporary input file based upon the input file name and the number of points (see Section 3.5.3). At the end of every run, the convergence runner will read the maximum outflow and maximum reservoir water surface elevation from the routing table in the listing and compare these values to those of the prior run using the limits. Drawdown runs will also compare the elevation at the reference time. Once a run meets the limits, the convergence runner will copy the `.OUT`, `.BRE`, `.BRI`, and `.HCT` files over to the projects output files.

When there is no output from the run loop (such as when 10 runs have been completed but none have met the criteria), the runner will use the last run that generated a routing table and inform the user what time increment the results were based on. All of the messages from all of the input files will be displayed together after the run (see Section 3.5.4). Chapter 4 evaluates how the changes made to the simulator affected its output.

3.7.2 Hydrograph resampling

The 2025 version of the main hydrograph screen did not have the capability to convert a hydrograph with a variable time increment into a hydrograph with a constant time increment; the conversion button only changed the table that was displayed. When a new constant time increment was entered, it rewrote the time column and left the discharge column blank.

We have added resampling capabilities to both the main hydrograph screen and the three

EFH-2 hydrograph screens (Figure B.4). Now when you convert a variable-time table to a constant time increment, or change the increment of a constant-time table, the discharges will be interpolated linearly onto the new increment and any negative values will be set to zero, just as they were in the legacy conversion routine. The legacy conversion routine allows for up to 17,000 points; ours allows for 5,000. If a hydrograph requires more than 5,000 points, we will automatically increase the increment until 5,000 points are sufficient to represent the entire duration of the hydrograph. A converted table will begin at time zero, and the first time of the variable-time table will become the start time of the hydrograph.

Conversion is required for the time-increment adjustment described in Section 3.7.1, which is applicable only to hydrographs with a constant time increment; a variable-time hydrograph is simulated only once and the user is informed why.

3.7.3 Results graphs

The *View Graphs* menu item for selected dam in the summary screen did not have any functionality in the 2025 version. We have now included 19 graphs of a run's results, all of which are listed in that menu (Figure B.5): Hydrographs; Reservoir Water Surface, Reservoir Surface Area, and Reservoir Storage Volume; Dam Crest Profile and Dam Cross-Section; Principal Spillway Rating, Auxiliary Spillway Ratings, Tailwater Rating, and All Discharge Ratings; Tailwater Elevation; Max Overtopping/Breach Discharge, Maximum Overtopping Head, and Overtopping Stress; and Headcut/Breach Width, Conduit/Breach Width, Headcut Position, Conduit Width/Height, and Headcut Advance (Section 3.7.4). Each graph is generated using OxyPlot on the summary-table page for the run selected on that page from the files associated with that run in the project directory: the simulator input (`.WDC`), the listing (`.OUT`), and the breach, internal-erosion, and breach-progression files (`.BRE`, `.BRI`, `.HCT`). The menu displays the conduit graphs for an internal-erosion run and the headcut graphs for other runs.

3.7.4 Headcut advance visualization

The 2025 version did not have a headcut advance graph. We have now provided a Headcut Advance graph to show how the breach will develop as the simulation runs (Figure B.6). This graph uses data generated by the Dam Simulator for the selected simulation: the breach file (`.BRE`) and the breach progression file (`.HCT`) if it is an overtopping breach; or the internal erosion file (`.BRI`). If no headcut occurs during the simulation, then the graph will display "No Headcut Formed."

The Headcut Advance graph displays one time increment in two panels. The top panel is a side view of the dam. The dashed line represents the dam prior to erosion. The solid red line represents the eroded dam with the headcut. The solid blue line represents the water surface. The bottom panel is a plan view of either the breach opening or, if it is internal erosion, of the conduit. You can use a slider to select which time increment you want to view. A label will indicate what time that time increment is.

We have also added a Play/Pause button. When you click Play, the slider will advance every 80 milliseconds. Since each run is limited to 600 frames, longer runs will skip some steps. When you click Pause, the playback will stop at the current frame. When you click Playback, the playback will continue until the last frame is reached and then begin playing again from the first frame.

3.7.5 Summary table field selection

The "Specify Summary Fields" option in the 2025 version did not have any function. We created a dialog box that opens up after selecting this option (Figure B.7). This dialog allows users to choose which fields will be displayed in their summary tables; there are 49 fields available for the dam-level table and 35 for the auxiliary-spillway table. All fields are selected by default. When the user makes their selections, those choices are saved in the user's application data folder and therefore apply to all projects. If a summary table is currently open, it will be rebuilt upon closing the dialog box using OK.

3.7.6 Summary graphs across runs

In the 2025 version the Summary Graphs button on the summary table page was inactive. We have now created a graph that allows you to compare runs (Figures B.8 and B.9). You select two summary fields (from those listed in the summary table) to use as the x-axis and y-axis. Then you can choose whether to create a scatter plot or a line graph. Each run in the table will generate one point on the graph. If a run has "N/A" for either field it is excluded. If a run has "BREACH" it is plotted as zero. If you create a line graph, the points are connected in order of their x-axis values. When you click "Summary Graphs", the graph will replace the summary table on the summary-table page. There are also buttons to allow you to change your selections and to switch between scatter plots and line graphs.

3.7.7 Latin hypercube sampling pipeline

In the 2025 version the summary table could read existing sample files, but the application could not produce them.

We have added a pipeline which will run after a simulation is complete if you choose to configure your auxiliary spillway for sampling (Figure 3.5). This configuration is saved as part of your project (Section 3.4.1) and includes the number of samples to take, a random seed, whether or not you want to use Latin Hypercube Sampling, and for each variable property of the spillway's soil materials, a distribution and its parameters (Figure B.10). The properties that can be varied include: Plasticity Index; Representative Diameter; Percent Clay; Detachment Rate; Dry Density; Headcut Index. For each spillway with at least one variable property, the pipeline creates the input for `DakotaLHS64`, a sampling program created by another contributor, runs it, reads the sample values it generates, and then creates one `.D2D` file for each sample, which is a copy of the spillway's input file (Section 3.5.2) with the sampled values in its material records. Finally, it runs SitesSim on all `.D2D` files in the run's subfolder, four at a time.

The summary table displays three results when a spillway has multiple results; these include the sample that had the most erosion, the sample that had the least erosion, and

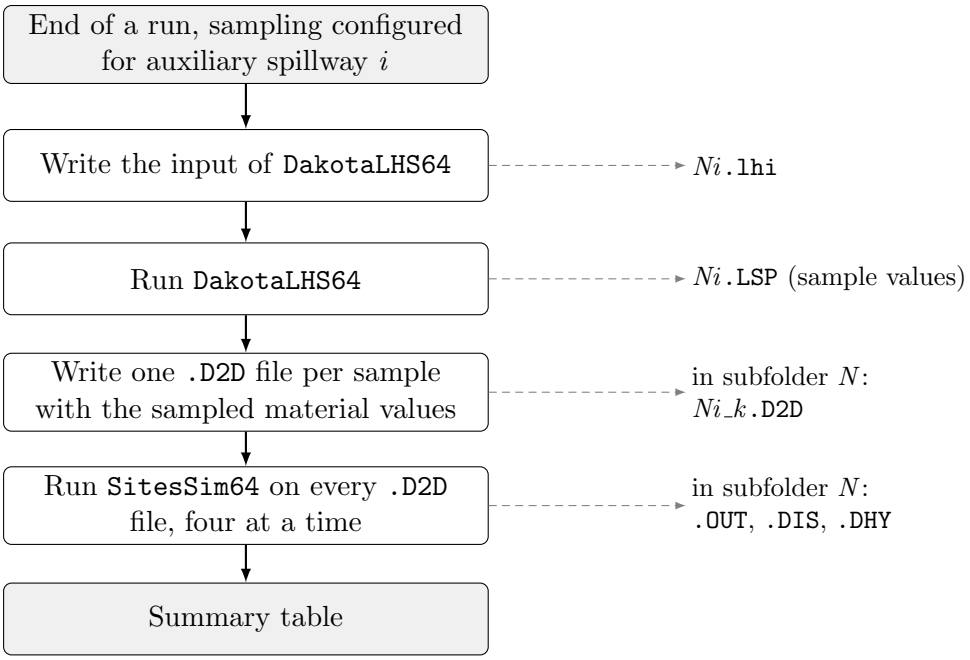


Figure 3.5: *Steps of the sampling pipeline for auxiliary spillway i, with the files each step writes. N is the name of the project file and k the number of a sample.*

the sample that was closest to the average. This display was available in the 2025 version. Dakota's Latin Hypercube routines were used by the legacy application to generate samples. Users were able to select Monte Carlo or Latin Hypercube sampling and provide a random seed.[13] The legacy application's summary table listed the run that had the most erosion, the run that was closest to the average, and the run that had the least erosion.[13]

Neilsen and Cao had also linked WinDAM C and the National Weather Service BREACH model with Dakota. This allowed both models to be run across various ranges of material and geometric inputs.[24]

3.8 Development and Physical Views: Build and Deployment

Every app is a Visual Studio solution containing two projects: the app itself and a WiX installer. In Table 3.3, you will find all of the main dependencies for each app and what the installers install onto your computer, as of October 2026. The WinDAM C installer was

not altered during our research; Burugupalli has detailed information about it.[16] We have detailed our changes to the EFH-2 build and installer in Section 3.6.

Table 3.3: *Build and installation of WinDAM C and EFH-2 as of October 2026. Folder names are those defined by the installers.*

	WinDAM C	EFH-2
Version	1.1.17	2.0.15.1
Platform	.NET 10; WinUI 3 with Windows App SDK 1.7; unpackaged	.NET 10; WinUI 3 with Windows App SDK 1.7; unpackaged
Main libraries	Newtonsoft.Json 13.0.3; OxyPlot 2.2.0 (H.OxyPlot.WinUI 0.9.30); CommunityToolkit.Mvvm 8.4.0	CommunityToolkit.Mvvm 8.4.0; H.OxyPlot.WinUI 0.9.30; PDFsharp-MigraDoc 6.1.1
Programs copied with the application	`WinDamSim64.exe`, `SitesSim64.exe`, `WinTR20_V64.exe`	`WinTR20_V64.exe`
Installer	WiX Toolset SDK 6.0.0	WiX Toolset SDK 5.0.2
Program folder	Program Files\USDA\ WinDAMv1.1.17	Program Files\USDA\EFH-2
Data folder	ProgramData\USDA\Shared Engineering Data\WinDAMC	ProgramData\USDA\Shared Engineering Data\EFH2
Start menu	Engineering Applications\ WinDAM C (64-bit)	Engineering Applications\ EFH-2

3.9 Summary

The WinDAM C and EFH-2 applications are two different software programs that communicate with each other using files. We have implemented several new features in WinDAM C including; Time-Increment Adjustment & Per-Screen Settings, Hydrograph Resampling, Results & Summary Graphs, Summary-Field Selection, and Sampling Pipeline (see Section 3.7). We have also modified the data, screens, platform, and installer within EFH-2 (see Section 3.6). Chapter 4 will test those aspects of our design which will allow us to reproduce the legacy results: The .WDC files created by the application (see Section 3.4), the output from the simulator when given identical input files (see Section 3.5), and the impact of the Time-Increment Adjustment (see Section 3.7.1).

Chapter 4

Validation and Results

This chapter checks whether the new 64-bit WinDAM C application as described in Chapter 3 produces the same results as the legacy 32-bit application. The same seven examples were run with the same input files in the new 64-bit WinDAM C application and in the legacy 32-bit application. Section 4.1 describes the cases and how they were run and compared, and Section 4.2 checks the input files. The results without automatic time-increment adjustment are compared in Section 4.3 and with automatic time-increment adjustment in Section 4.4. Results for sampled inputs are compared in Section 4.5. Section 4.6 summarizes the differences found.

4.1 Test Design

4.1.1 Test Cases

We used seven cases of the WinDAM C examples as test cases. Each case is named by its input file (Table 4.1). Two of the cases, `05-HR-OvertopBreach` and `06-TH-OvertopBreach`, model embankments that overtop and then breach; `05-HR-OvertopBreach` uses the Hanson/Robinson headcut model and `06-TH-OvertopBreach` uses the Temple/Hanson model. In `04-HR-Overtop`, the embankment is overtopped but does not breach. Cases `10-FBIntegrity2AuxSpwy` and `11-FBIntegrity2AuxSpwyTailwater` model erosion in two

auxiliary spillways that are not overtopped, with the second case including a tailwater rating. Cases `14-IE-medKd050` and `15-IE-fastKd500` model internal erosion with detachment rate coefficients k_d of 0.50 and 5.00 ft/h/psf, respectively.

The two overtopping-breach cases were run with an inflow hydrograph with a constant time increment of 0.02 h (1491 values), while the other five cases were run with a variable time increment (185 points). All cases were run with the automatic time-increment adjustment (Section 3.7.1) off and on. `04-HR-Overtop` was run only with the adjustment on, because its original file already has it on. There are 13 runs per application, so 26 runs in total. The 64-bit application only applies the adjustment to hydrographs with a constant time increment (Section 3.7.2).

4.1.2 Programs

Table 4.2 lists the programs used for the runs. The legacy program is the installed 32-bit WinDAM C version 1.1.15, which NRCS released in August 2024.[9] The 64-bit program is the application described in Chapter 3, built in October 2026; Appendix A gives its build record. In each program the user interface prepares the input files and runs the dam simulator and the spillway simulator, which do the calculations; the simulators of both programs are version 1.1.15, but they are separate 32-bit and 64-bit executables (Table 4.2). The four simulator programs are those stored with the Fortran source in the repository, whose build scripts compile the same source once as 32-bit and once as 64-bit programs; for the spillway simulator the two scripts also use different optimization settings.

4.1.3 Running the Cases

The runs with the adjustment on were done by opening each original file in the legacy program, turning the option on, and then saving the file with a new name. The only differences between the original files and the new files are the HYD record (which holds the time increment of the inflow hydrograph and the adjustment setting as explained in Section 3.4.1) and the lines that record where and when the file was saved. Note that for

Table 4.1: *Test cases, named by input file. Later tables use the number prefix. The description is that of the legacy run with the adjustment off (`04-HR-Overtop`: on); headcut models are those printed by the dam simulator. All cases were run with the adjustment off and on, except `04-HR-Overtop` (on only).*

File name	What it tests	Inflow hydrograph
`04-HR-Overtop`	Overtopping without breach; Hanson/Robinson headcut model; one auxiliary spillway	Variable, 185 points
`05-HR-OvertopBreach`	Overtopping breach; Hanson/Robinson headcut model; one auxiliary spillway	Constant, 0.02 h
`06-TH-OvertopBreach`	Overtopping breach; Temple/Hanson headcut model; one auxiliary spillway	Constant, 0.02 h
`10-FBIntegrity2AuxSpwy`	Not overtopped; erosion in two auxiliary spillways	Variable, 185 points
`11-FBIntegrity2AuxSpwyTailwater`	As 10, with a tailwater rating	Variable, 185 points
`14-IE-medKd050`	Internal erosion, k_d = 0.50 ft/h/psf; no auxiliary spillway	Variable, 185 points
`15-IE-fastKd500`	Internal erosion, k_d = 5.00 ft/h/psf; no auxiliary spillway	Variable, 185 points

Table 4.2: *Programs used for the runs. Simulator versions are those printed in their listings.*

Program	Component	Executable	Version
Legacy	User interface	`WinDAMC.exe`	1.1.15
Legacy	Dam simulator	`WinDamSim.exe`	1.1.15
Legacy	Spillway simulator	`SitesSim.exe`	1.1.15
64-bit	User interface	`WinDAMC.exe`	1.1.17 (development build, October 2026)
64-bit	Dam simulator	`WinDamSim64.exe`	1.1.15
64-bit	Spillway simulator	`SitesSim64.exe`	1.1.15

04-HR-Overtop the limits of its original file (2 percent and 0.01 ft) were kept.

All runs with the adjustment were done through the application's interface. This means that for each run we copied the input file into an empty folder, opened the file in the program and then ran the simulation without changing any of the input. All files written

by the program were kept. Both programs ran identical input files (checked file by file). All runs finished and each main listing ends with "End of Run". All runs were made on one computer running Windows 11.

4.1.4 Quantities Compared

To compare the runs we first compared the key results that the simulators report such as the peak outflow, the maximum pool elevation, the breach times and the spillway erosion. Then we compared the results over time, row by row: the routing table as listed by the dam simulator, i.e. the inflow, the outflow and the water level at each time step, and the headcut position. Finally, we compared every output file present in both runs with numdiff (Section 4.1.5). The comparison of all output files between the two runs is done with the tool numdiff (Section 4.1.5). The difference is always calculated as the run under test minus the reference run (the legacy run in case of comparison between the two programs and the run with the adjustment turned off in case of measuring the effect of the adjustment).

4.1.5 Agreement Criteria

We compared files using numdiff 5.9.0. We used the tolerances of Burugupalli, 0.05 relative and 0.01 absolute, and her way of calling numdiff.[16] A number agrees with a reference value when it is within 5 percent of the reference value or within 0.01 of the reference value. Two files agree when no number in the files is outside the tolerance and the two files have the same number of lines.

All output files of both programs end with a carriage return (CR) which is not treated as a separator by numdiff by default. Thus, the last number on each line would be treated as text. We included CR as a separator in our numdiff command. Appendix A contains the full command. If only text differs between two files, we report that the numbers match and the text differs. For the key results we report the largest actual difference between the files, rather than a simple pass or fail.

4.2 Input Files

Section 4.1.3 already showed that both programs processed the same input files. Now we would like to know whether the 64-bit program also writes the same input file when saving a project. We opened each of the seven input files with the automatic time-increment adjustment (Section 3.7.1) on in the 64-bit program, saved it without changing anything, and compared it line by line with the legacy-written original, leaving out the lines that record where and when the file was saved.

All saved files are different from the original files. The number of different lines ranges between 12 and 100 per file. Most of the differences are of a formatting nature (e.g. 59.20 written as 59.2, .013 written as 0.013, indentation of a comment line changed, etc.). However, in all cases three content changes were found: the date of creation on the first line of the file was changed from 03/07/2024 to 10/06/2022 (this is a fixed date written by the 64-bit program); the keyword `OTHER` was added to the `OPTION` record; an empty comment line was added.

For the two internal-erosion cases, saving the project with the 64-bit program even loses the internal-erosion setup, i.e. the IEMODEL record with the erosion parameters is written as an HCMODEL record, and the CONDUIT record with the size and position of the conduit is missing. The 64-bit program reads both records, but does not write them (Section 3.4.3). These changes did not affect the results presented in this chapter, because all runs were performed with the original input files.

4.3 Results at the Same Time Increment

With the automatic time-increment adjustment (Section 3.7.1) off, the two programs agree closely: peak outflow is within 0.1 percent in six of seven cases (Table 4.3). The exception is `05-HR-OvertopBreach`, where the 64-bit peak is 0.594 percent lower; in `10-FBIntegrity2AuxSpwy` the two peaks are the same. Among the other key results, the largest difference is also in `05-HR-OvertopBreach`: 0.551 percent in the maximum discharge

through the breach, which is complete 0.04 h earlier in the 64-bit run. The only other difference in time is in `14-IE-medKd050`, where peak outflow occurs one time increment, 0.162 h, later.

There are two points to make here: 1) `04-HR-Overtop` does not have a "no run" comparison for the adjustment on/off, so we compared the first simulator run of the legacy adjustment (185 points from the input hydrograph) with the single 64-bit run. 2) For the five variable-time cases, the legacy program began the inflow at 0 cfs instead of the 0.83 cfs in the input file; the 64-bit program used the input as given.

Table 4.4 lists the 75 output files compared with numdiff (Section 4.1.5) in increasing order of why they differ. Thirty-two files agree. Most other files differ either because the legacy program starts the inflow of the variable-time cases at 0 cfs, because the breach rise of `05-HR-OvertopBreach` is a little ahead in the 64-bit run, or because the 64-bit program labels the files of the second auxiliary spillway with the name of the first auxiliary spillway.

The results for the routing tables of the dam simulator are very similar to each other. The largest difference in total outflow at one time is less than 0.4 cfs in `04-HR-Overtop`, `10-FBIntegrity2AuxSpwy` and `11-FBIntegrity2AuxSpwyTailwater`. For the variable-time cases, the results for times after 32.236 h are not compared, because the printed times of the two programs start to diverge after that point. The difference in total outflow for the remaining times is between 35.74 and 80.66 cfs (0.195 to 1.97% of the legacy value) in `06-TH-OvertopBreach`, `14-IE-medKd050` and `15-IE-fastKd500`.

The exception is `05-HR-OvertopBreach`, where one row differs by 1,595.07 cfs (5.72 percent). This is a shift in time: during the breach rise, when the outflow grows by more than 900 cfs every 0.02 h, the 64-bit run is 0.02 to 0.04 h ahead, and its breach is complete at 16.37 h instead of 16.41 h.

4.3.1 Erosion Over Time

Figures 4.1 and 4.2 are the outflow and the headcut for the run `06-TH-OvertopBreach` with the adjustment off. The legacy run is drawn as a line and the 64-bit run as markers. The

Table 4.3: *Peak outflow with the adjustment off; cases by file-name prefix (Table 4.1). For 04, the first legacy simulator run and the single 64-bit run.*

Case	Legacy (cfs)	64-bit (cfs)	Difference (%)
04	2,707.01	2,706.95	0.00222
05	33,667.41	33,467.35	0.594
06	32,629.56	32,618.63	0.0335
10	3,377.81	3,377.81	0
11	3,367.49	3,367.64	0.00445
14	8,243.60	8,235.53	0.0979
15	9,077.22	9,076.95	0.00297

Table 4.4: *Output files compared with numdiff, adjustment off, by the reason they differ. "Agree" includes files in which only text such as dates and paths differs.*

Reason	Files
Agree	32
Different first inflow (variable-time cases)	17
Breach rise ahead in `05-HR-OvertopBreach`	7
Second spillway labelled as the first (64-bit)	8
Format only (one extra end line)	2
Other (extra routing row, warning, flow switch, graph end)	9
Total	75

lower panel of Figure 4.1 is the 64-bit value minus the legacy value. The outflow is the total outflow in the routing table of the listing for the dam simulator (`.OUT`), and the headcut is the headcut station in the breach-progression file (`.HCT`; Table 3.2). Both programs are on top of each other. The largest difference in outflow at one time is 50.08 cfs (0.195 percent of the legacy value) at 17.71 h, while the breach is opening. In the summary of both listings, the breach starts at 16.51 h and is complete at 17.85 h. The headcut position differs by at most 0.11 ft. It reaches station 0 ft, the upstream end of the embankment, where the dam simulator treats the breach as complete, at 17.85 h in both programs.

The spillway simulator writes the eroded surface of the auxiliary spillway to the `.DG6` file only at the end of the run. Thus, for the spillways only the final surface can be compared. For the 17 points in `10-FBIntegrity2AuxSpwy` for both spillways, the eroded surfaces are identical in both programs.

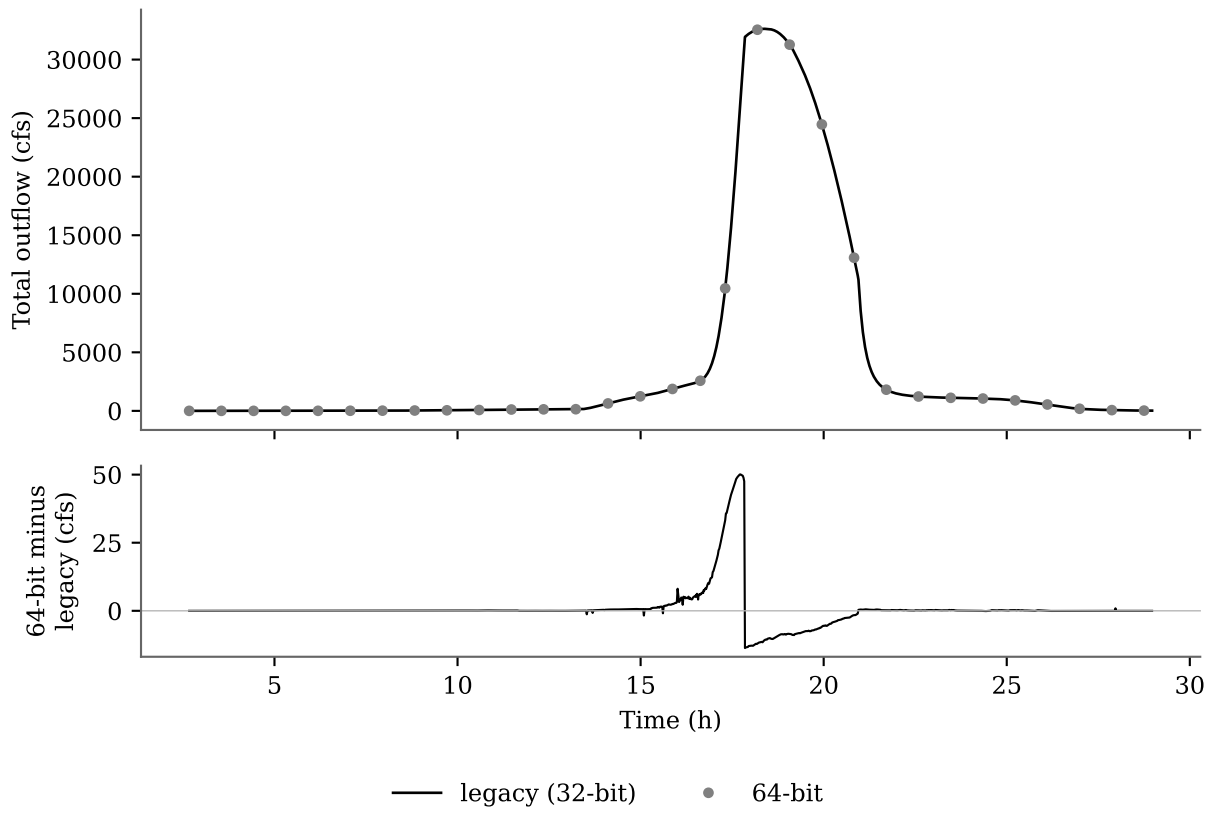


Figure 4.1: *Total outflow of* `06-TH-OvertopBreach` *with the adjustment off: legacy (line) and 64-bit (markers); below, 64-bit minus legacy.*

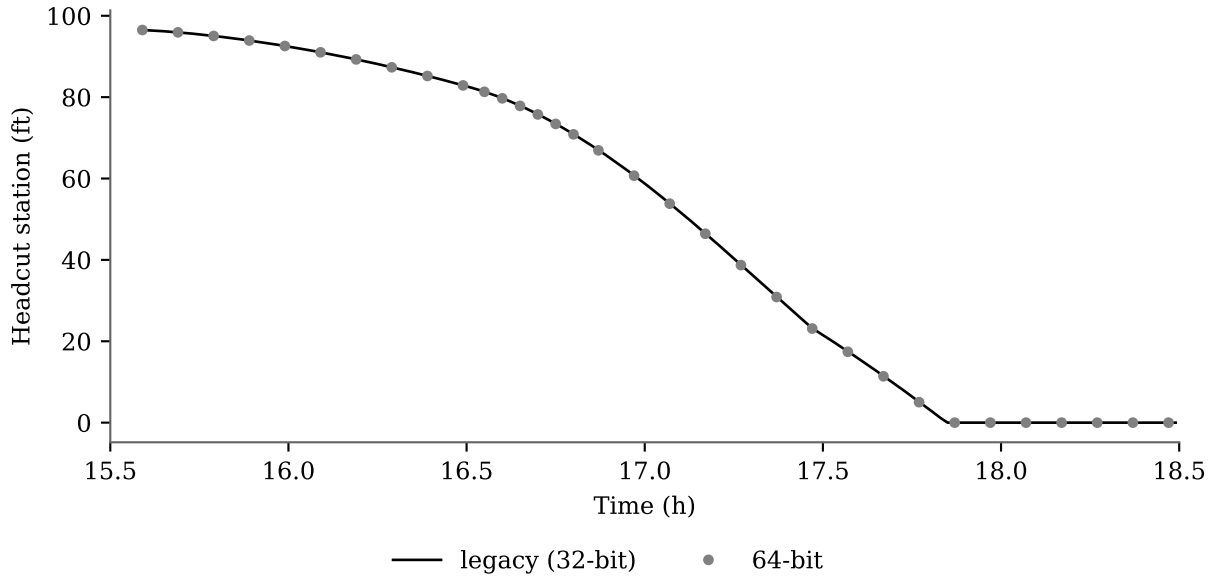


Figure 4.2: *Headcut station of* `06-TH-OvertopBreach` *with the adjustment off, from 15.5 to 18.5 h: legacy (line) and 64-bit (markers). After 17.85 h it stays at 0 ft in both programs.*

In the internal-erosion case `14-IE-medKd050` the differences in the routing table (max 80.66 cfs, 1.97% error in single rows) occur when the two programs switch between the two different types of conduit flow one time step apart. The kind of flow and the position of the conduit top come from the internal-erosion file (`.BRI`); the conduit top also reaches the crest one time increment later in the 64-bit run (15.39 h instead of 15.23 h)

4.4 Automatic Time-Increment Adjustment

With the automatic time-increment adjustment (Section 3.7.1) on, the application reruns the dam simulator with half the time increment until peak outflow and peak water-surface elevation change by less than the limits on the `HYD` record. Table 4.5 shows where it acted. In the two constant-time cases both programs made the same two runs, first with 1491 points at 0.02 h and then with 2981 points at 0.01 h, and kept the second.

In the five variable-time cases the legacy program also made two runs, the second with each time interval of the inflow split in two (185 to 369 points). The 64-bit application ran each of these cases once, because it does not apply the adjustment to hydrographs with a variable time increment (Section 3.7.2).

Table 4.6 compares the two programs with the adjustment off and on in the two constant-time cases. It shows the main eight of the key quantities, out of 15 compared for `05-HR-OvertopBreach` and 18 for `06-TH-OvertopBreach`. With the adjustment on, legacy and 64-bit agree as well as or better than without it for nearly all key quantities (17 of 18 in `06-TH-OvertopBreach`, 13 of 15 in `05-HR-OvertopBreach`), and peak outflow agreement improves (0.594 percent $\rightarrow$ 0.0316 percent in `05-HR-OvertopBreach`). The three exceptions are the maximum flow over the dam and the time of peak outflow in `05-HR-OvertopBreach`, and the time of the ending low point in `06-TH-OvertopBreach`.

Figure 4.3 shows the outflow of both programs in `06-TH-OvertopBreach`, with the adjustment off and on; the lower panels show the 64-bit value minus the legacy value. The largest difference at one time falls from 50.08 cfs (0.195 percent of the legacy value, while the breach opens) to 15.93 cfs (0.144 percent, at 20.94 h) when the adjustment is on. In

Table 4.5: *Simulator runs made with the adjustment on; cases by file-name prefix (Table 4.1).*

Cases	Inflow hydrograph	Legacy	64-bit
05, 06	Constant, 0.02 h	Two runs: 1491 points, then 2981 points at 0.01 h; second kept	The same
04, 10, 11, 14, 15	Variable	Two runs: 185 points, then 369 points (intervals split in two); second kept	One run; adjustment not applied

Table 4.6: *Legacy against 64-bit with the adjustment off and on, cases 05 and 06 (file-name prefixes, Table 4.1): size of the difference in percent of the legacy value, or 64-bit minus legacy in hours.*

		05		06	
Quantity	Unit	Off	On	Off	On
Peak outflow	%	0.594	0.0316	0.0335	0.004
Max. auxiliary-spillway flow	%	0.0198	0.0185	0.0356	0.00636
Max. flow over the top of the dam	%	0.208	0.757	0.299	0.0714
Max. breach width	%	0.165	0.00827	0.00867	0
Max. discharge through the breach	%	0.551	0.029	0.0312	0.00294
Time of peak outflow	h	0.00	−0.01	0.00	0.00
Breach complete	h	−0.04	0.00	0.00	0.00
Time of ending low point	h	0.00	0.00	0.00	−0.01

`05-HR-OvertopBreach` it falls from 1,595.07 to 87.14 cfs (0.307 percent, at 16.28 h).

With the adjustment on, 9 of the 13 output files agree in `05-HR-OvertopBreach`: the two spillway files differ only in format, and the breach and breach-progression files differ in 14 and 4 values. In `06-TH-OvertopBreach` 8 of the 13 agree; the other five have a different number of lines, because the dam simulator's files have one or two rows more or fewer, or because of the format of the spillway files.

The adjustment also changes each program's own results, by up to 4.26 percent (the maximum flow over the dam in `05-HR-OvertopBreach`, legacy). In that case the 64-bit peak outflow changes more than the legacy one (0.657 against 0.027 percent), so that with the adjustment on the two peaks differ by only 0.0316 percent. These runs compare the two programs at a smaller time increment; they do not show which program is right.

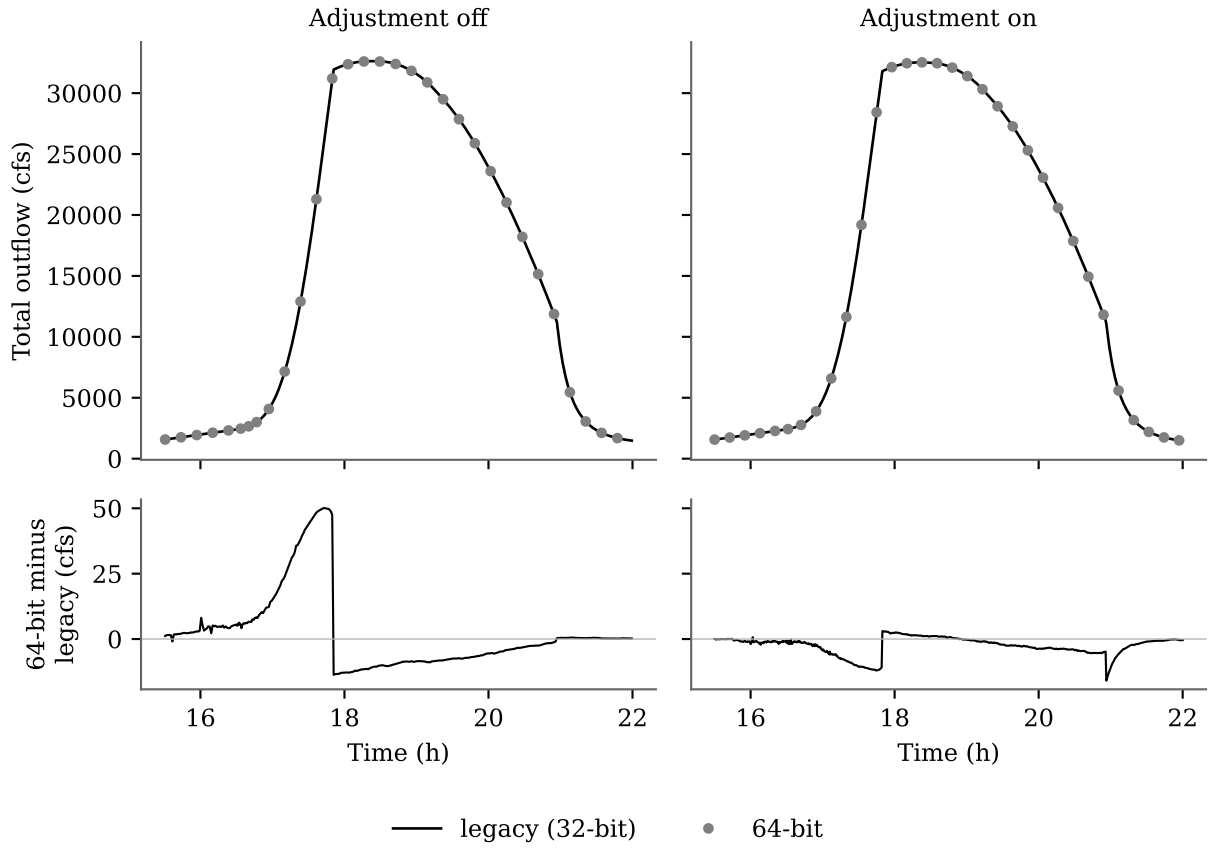


Figure 4.3: *Total outflow of* `06-TH-OvertopBreach` *from 15.5 to 22 h in the legacy (line) and 64-bit (markers) programs, with the adjustment off (left) and on (right); below, 64-bit minus legacy.*

In the 5 variable time cases nothing changes in the 64-bit program when the setting is changed. The outputs of the program with setting on and off are identical except for the lines of output that are echoed in the run. The application of the setting to hydrographs with variable time increment in the 64-bit program is left as future work (Chapter 5).

4.5 Sampling-Based Comparison

Latin hypercube sampling (LHS) is a sampling method that covers the sample space more evenly than random sampling. In order to sample N values of a uniform range, the range is divided into N equal intervals and one value is sampled at random from each interval. In both programs, the material properties of the auxiliary spillway are sampled using LHS

(Section 3.7.7). The steps are the same in both programs. The project file contains a record for LHS sampling in the form of a `LHS` record. After the dam simulator has run, the sampling settings are written to a sampling input file (*.lhi) and the sampling program is run, which writes the sampled values to a *.LSP file. The program then writes one spillway input file (*.D2D) for each sample, with the sampled value in the material record, and runs the spillway simulator on each file.

We used `06-TH-OvertopBreach` for both programs with the automatic time-increment adjustment (Section 3.7.1) turned off. All other settings were identical between the two programs. The first spillway material had a plasticity index of 12 in the input file, which was varied uniformly between 6 and 24 with 10 samples, seed 12345, and Latin hypercube sampling. Each program saved the project before running. The two records below were added to the project file; they give the number of samples, the seed, and the sampling options, followed by the material, the property, the distribution, and its limits. Below them is the material record of the first sample.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
LHS        10         12345     1         1
3    1     1          10                  6          24
ENDTABLE
           1          12         .008      20         110        .07      input
           1          14.9684117.008      20         110        .07      legacy
           1          14.9684    .008      20         110        .07      64-bit
```

Table 4.7 compares the files that the two programs wrote.

The two programs generated the same ten values in the same order. Each of the ten values was generated in each of the ten tenths of the range of values for the sample. The values generated by the two programs differ by at most 0.000002. This difference is due to the use of different versions of the sampling program (2.30 and 2.20).

The results for the spillway agree well with each other. For the flow duration, for instance, a value of 5.4 h is found in every sample. For the headcut formation, the same seven samples are found in both programs to form a headcut. All of them are found to form the headcut at station 375 ft where the headcut started. The final headcut height for the samples that

Table 4.7: *Files written by the legacy and 64-bit programs in the sampling run of* `06-TH-OvertopBreach`.

File	Same in both?	Note
Sampling records in the project file (`.WDC`)	Yes	The 64-bit save also makes the changes of Section 4.2
Sampling input (`.lhi`)	Yes	Apart from the file paths
Samples (`.LSP`)	Same values	Differ by at most 0.000002; printed with 15 and 9 digits
Dam results (`.OUT`, `.BRE`, `.HCT`)	As in Section 4.3	Not changed by the sampling
Spillway input per sample (`.D2D`)	No	Sampled value written differently; inflow from the dam run
Spillway files per sample	Same set	Ten samples, eight files each, in a subfolder

form a headcut, ranges between 0.65 ft and 8.55 ft. The difference between the results of the two programs for the final headcut height for these samples is at most 0.01 ft, i.e. 7.05 ft against 7.04 ft. The difference between the results of the two programs for the eroded area for these samples is at most 0.95 ft^2 (0.095 percent). The difference between the results of the two programs for the peak discharge for all samples is 0.56 cfs (0.036 percent). This is the inflow from the dam run, which is the same as in Section 4.3.

The legacy program pads out values to 10 characters or truncates them as required. All values written by the legacy program in the material record are written to fill the 10-column fields in which they are placed, and then spill over into the next field (see excerpt above). The spillway simulator reads material records in fixed 10-column fields and outputs the representative diameter of 0.008 in and plasticity index rounded to the whole number of the sample for each program run. This program reads the intended values. The 64-bit program rounds values to four or five decimals and then pads them out to 10 characters. The amount of change in the sampled values is less than 0.00005, and no values are written to the wrong fields by either program.

4.6 Differences Found

The user interface for the legacy program prepares the input files for the simulators. Inflow of a variable-time hydrograph begins at 0 cfs rather than at 0.83 cfs as indicated in the input file. The automatic time-increment adjustment (Section 3.7.1) is applied only by the legacy program for such hydrographs. The 64-bit program fails to save the internal-erosion setup for a project and designates the second auxiliary spillway as the first in the saved project. Both of these 64-bit program errors were present when work began on these runs and did not affect them.

There are some differences between the two sets of simulator results. The legacy spillway summary reports a breach of the spillway that is not listed in the summary. The 64-bit program reports numbers for the same summary lines that are correct only by chance. These summary lines have been omitted from the comparison between the two sets of results. The breach in the 64-bit run for `05-HR-OvertopBreach` was higher in some instances than in the 32-bit run, in some cases up to 5.72 percent higher in some rows. In `06-TH-OvertopBreach` some single rows were higher in the 64-bit run by up to 50.08 cfs, and the legacy routing table had one more row than the routing table generated by the 64-bit program. For both cases the dam simulator reads the same input file, and no other files are read by the dam simulator. These differences can therefore be attributed to the two dam-simulator executables (32-bit and 64-bit) of the same source version (see Section 4.1.2).

Running the two dam-simulator executables on the same input files (no user-interface) gave the same results for each of the two programs. The part of the build that was responsible for the differences was not isolated. In `14-IE-medKd050` the same flow switch happens one time increment apart in the two programs, which changes single rows by up to 1.97 percent. Inflow for this case also differs between the programs and so its cause was not traced. Other small differences are not explained; the rest are in output format only and change no value.

These results are limited. They consist of seven example cases run on one computer and agreement means the same result as the legacy program. The adjustment was compared for two cases and the sampling for one case and one property. The effect of the different first

inflow was not measured.

4.7 Summary

The legacy 32-bit WinDAM C program was compared to the new 64-bit application in seven cases with the automatic time-increment adjustment turned off and on and in one sampling run. With the adjustment turned off, peak outflow from the seven cases agreed within 0.1 percent in six of the cases (Section 4.3). The only exception was `05-HR-OvertopBreach` where the breach started to rise slightly faster in the 64-bit simulation run than in the 32-bit simulation run. With the adjustment turned on, the two programs agreed as well as or better than when the adjustment was turned off for almost all of the key quantities (17 of 18 in `06-TH-OvertopBreach`, 13 of 15 in `05-HR-OvertopBreach`). For the one exception, the difference in peak outflow between the two programs was reduced to 0.0316 percent (Section 4.4). The 64-bit program does not yet apply the adjustment to hydrographs with a variable time increment. In the lhs run, both programs produced the same ten values, the spillway results also agreed closely, the final headcut heights were within 0.01 ft, and no values were written into the wrong field by either of the program (Section 4.5).

The differences found between the legacy program and the 64-bit program include the first inflow of 0 cfs in the variable-time cases, the loss of the internal-erosion setup when saving a project with the 64-bit program, the labeling of the second spillway in the 64-bit program, and the summary of the legacy spillway that reports a breach that did not occur (Section 4.6). Future work is described in Chapter 5.

Chapter 5

Conclusions and Future Work

This chapter answers the research question of Section 1.3 from the work of Chapter 3 and the comparison of Chapter 4, states the limits of that comparison, and lists the work that remains.

5.1 Conclusions

We enhanced two software applications. In the standalone EFH-2 application, we exchanged the soils database, created five new rainfall distribution databases, modified five screens in the application and the help, converted the application to run on .NET 10, and upgraded its installer to version 2.0.15.1 (Section 3.6). For the 64-bit WinDAM C application of 2025, (Section 2.5), we incorporated an automated time increment adjustment (Section 3.7.1) along with convergence options for each screen; we also included hydrograph resampling, 19 graphs representing a single run's results, including a headcut advance graph, multiple field options for the summary table, summary graphs across runs, and a Latin Hypercube Sampling Pipeline for the auxiliary spillway design (Section 3.7).

WinDAM C (64-bit) produces results that are nearly identical to those produced by the legacy 32-bit version of the program. The 64-bit version of WinDAM C produced results that were almost exactly the same as the legacy 32-bit version of the program when running

the same input file. When the adjustment was turned off, peak outflow differed by less than 0.1% in six out of seven test cases. The only test case that had a different result was `05-HR-OvertopBreach` which showed a difference of 0.594% (Section 4.3). When the adjustment was turned on, the two programs agreed as well as or better than they did without the adjustment for almost every quantity (17 out of 18 in `06-TH-OvertopBreach`, 13 out of 15 in `05-HR-OvertopBreach`). The peak outflow difference in `05-HR-OvertopBreach` was reduced to 0.0316% (Section 4.4). Both programs drew the same ten values in the sampling run. The programs differ in five ways (Section 4.6).

The 64-bit application does not adjust variable time increment hydrographs correctly and, because of two bugs that were present at the start of this work, loses the internal-erosionsetup when it saves a project, and lists the second auxiliary spillway as the first one. The legacy application starts inflow of variable-time hydrographs with 0 cfs instead of 0.83 cfs in the input file. The summary of spillways in the legacy application falsely reports a breach that did not occur.

5.2 Limitations

There are some limitations to the comparison in Chapter 4. The seven test cases were run on one computer and only agreement with the legacy program was tested. This means that the results are the same but not necessarily correct. The automatic time-increment adjustment (Section 3.7.1) was tested for two cases and the sampling for one case and one property. The effect of the different first inflow was not measured and the input files saved by the 64-bit program were never run (Section 4.2). The spillway results for the case `04-HR-Overtop` were not compared because the legacy program only keeps these results for the second run, which the 64-bit program does not do.

No results are reported for EFH-2. The modernized EFH-2 application was not compared with the legacy EFH-2 program, so this thesis does not show whether the two give the same results.

5.3 Future Work

The 64-bit solution has just the application and installer (see Section 3.8) and there are no tests to verify if you can read/write a `.WDC` file and get back the exact same file (see Section 3.4.3). There are three types of automated tests that would be useful. First, unit tests for both the reader/writer will verify each record in the input file. Second, round-trip testing will open/save all of the example projects and then compare them to their original versions. Lastly, regression testing will execute the example cases and then compare the output files to the reference outputs based upon the tolerance levels defined in Chapter 4 (see Section 4.1.5) so that any changes that produce different results will be identified prior to releasing.

The app will need to be easier for new users to get up-to-speed with. By refining the graphs and providing them with more appropriate tools to read and explore their results; this would reduce the amount of time it takes for a new user to become proficient.

Additionally, by integrating the app into the cloud, users can store their projects and results in a shared location, and large sampling studies can be performed on cloud computers rather than on a single desktop (Section 3.7.7).Lastly, testing needs to be expanded to include additional example cases, more sampled properties, and the EFH-2 app compared to the legacy EFH-2 program.

Bibliography

[1] Mitchell L. Neilsen, Darrel M. Temple, and Gregory J. Hanson. WinDAM: Earthen embankment erosion analysis. In *Proceedings of the 24th International Conference on Computers and their Applications in Industry and Engineering (CAINE 2011)*. International Society for Computers and Their Applications (ISCA), November 2011.

[2] Sherry L. Hunt, Darrel M. Temple, Mitchell L. Neilsen, Abdelfatah A. Ali, and Ronald D. Tejral. WinDAM C: Analysis tool for predicting breach erosion processes of embankment dams due to overtopping or internal erosion. *Applied Engineering in Agriculture*, 37(3):523–534, 2021. doi: 10.13031/aea.14334.

[3] USDA Natural Resources Conservation Service. Earth dams and reservoirs. Technical Release 210-60, USDA Natural Resources Conservation Service, Conservation Engineering Division, March 2019.

[4] USDA Natural Resources Conservation Service. *Chapter 21: Design Hydrographs*. USDA Natural Resources Conservation Service, May 2019. Part 630 Hydrology, National Engineering Handbook (210-VI-NEH, Amendment 87).

[5] USDA Natural Resources Conservation Service. Watershed rehabilitation program (REHAB). NRCS website, 2026. URL https://www.nrcs.usda.gov/programs-initiatives/watershed-rehabilitation-program. Accessed October 13, 2026.

[6] Robert B. Jansen. *Dams and Public Safety: A Water Resources Technical Publication*. U.S. Department of the Interior, Bureau of Reclamation, 1983. URL https://www.usbr.gov/tsc/techreferences/mands/mands-pdfs/AZ1130.pdf. Revised reprint.

[7] Francesca Aureli, Andrea Maranzoni, and Gabriella Petaccia. Review of historical dam-break events and laboratory tests on real topography for the validation of numerical models. *Water*, 13:1968, 2021. doi: 10.3390/w13141968.

[8] Tony L. Wahl. Prediction of embankment dam breach parameters: A literature review and needs assessment. Dam Safety Research Report DSO-98-004, U.S. Department of the Interior, Bureau of Reclamation, Dam Safety Office, Water Resources Research Laboratory, July 1998.

[9] USDA Natural Resources Conservation Service. WinDAM version 1.1.15. NRCS website, 2024. URL https://www.nrcs.usda.gov/resources/tech-tools/windam-version-1115. Version released August 2024. Accessed October 13, 2026.

[10] Tony L. Wahl. Uncertainty of predictions of embankment dam breach parameters. *Journal of Hydraulic Engineering*, 130(5):389–397, May 2004. doi: 10.1061/(ASCE)0733-9429(2004)130:5(389).

[11] Darrel M. Temple and Gregory J. Hanson. Headcut development in vegetated earth spillways. *Applied Engineering in Agriculture*, 10(5):677–682, 1994.

[12] Anthony Atkinson and Mitchell Neilsen. Analysis of uncertainty in internal erosion simulations for DLBreach and WinDAM C. *GeoHazards*, 5:350–363, April 2024. doi: 10.3390/geohazards5020018.

[13] Mitchell L. Neilsen and Chendi Cao. Sensitivity analysis of internal erosion models for dam safety. In *Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 457–464, Las Vegas, NV, USA, 2019. doi: 10.1109/CSCI49370.2019.00089.

[14] USDA Natural Resources Conservation Service. *Chapter 2: Estimating Runoff Volume and Peak Discharge*. USDA Natural Resources Conservation Service, 2nd edition, February 2021. Part 650 Engineering Field Handbook, National Engineering Handbook, Title 210 (210-650-H).

[15] USDA Natural Resources Conservation Service. ENG – WinTR-20 version 3.20.0000, Project Formulation – Hydrology computer program and user documentation. User Guide UG 210-21-5, USDA Natural Resources Conservation Service, April 2021.

[16] Samhitha Burugupalli. Modernizing legacy desktop applications: Re-engineering WinDAM C from VB6 to WinUI 3. Master's thesis, Department of Computer Science, Kansas State University, Manhattan, Kansas, 2025.

[17] Microsoft. Visual Basic 6.0 support announcement. Microsoft Lifecycle documentation, 2008. URL https://learn.microsoft.com/en-us/lifecycle/announcements/visual-basic-6-support-announcement. Originally published April 8, 2008. Accessed October 11, 2026.

[18] Santiago Comella-Dorda, Kurt Wallnau, Robert C. Seacord, and John Robert. A survey of legacy system modernization approaches. Technical Note CMU/SEI-2000-TN-003, Software Engineering Institute, Carnegie Mellon University, April 2000.

[19] Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, and Paul Wilson. Best practices for scientific computing. *PLoS Biology*, 12(1):e1001745, January 2014. doi: 10.1371/journal.pbio.1001745.

[20] William L. Oberkampf and Timothy G. Trucano. Verification and validation in computational fluid dynamics. Sandia Report SAND2002-0529, Sandia National Laboratories, Albuquerque, New Mexico and Livermore, California, March 2002.

[21] Philippe Kruchten. Architectural blueprints—the "4+1" view model of software architecture. *IEEE Software*, 12(6):42–50, November 1995.

[22] Microsoft. .NET and .NET Core support policy. .NET website, 2026. URL https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core. Last updated September 8, 2026. Accessed October 12, 2026.

[23] Microsoft. Process.StandardOutput property (System.Diagnostics). .NET API reference, Microsoft Learn, 2025. URL https://learn.microsoft.com/en-us/dotnet/api/system.diagnostics.process.standardoutput. Accessed October 12, 2026.

[24] Mitchell L. Neilsen and Chendi Cao. Coupled dam safety analysis using BREACH and WinDAM. In *Proceedings of the International Conference on Scientific Computing (CSC'17)*, pages 82–85. CSREA Press, 2017. ISBN 1-60132-452-9.

Appendix A

Validation Records

A.1 Programs Used in Chapter 4

The legacy program used for the runs of Chapter 4 is the installed release of WinDAM C version 1.1.15 (32-bit). The 64-bit program is version 1.1.17 of the application described in Chapter 3, built in October 2026 as a Release x64 build. The dam and spillway simulators of both programs print version 1.1.15.

A.2 File Comparison Command

Files were compared with numdiff 5.9.0 (run under Linux on the same computer) using the command

```
numdiff -E -F 2 -r 0.05 -a 0.01 -s $' \t\n\r' CANDIDATE REFERENCE
```

where `CANDIDATE` is the file being checked (the 64-bit run, or the run with the adjustment on) and `REFERENCE` is the file it is compared with (the legacy run, or the run with the adjustment off). The option `-s` makes the blank, tab, line feed and carriage return separate the fields of a line.

Appendix B

User Interface Screenshots

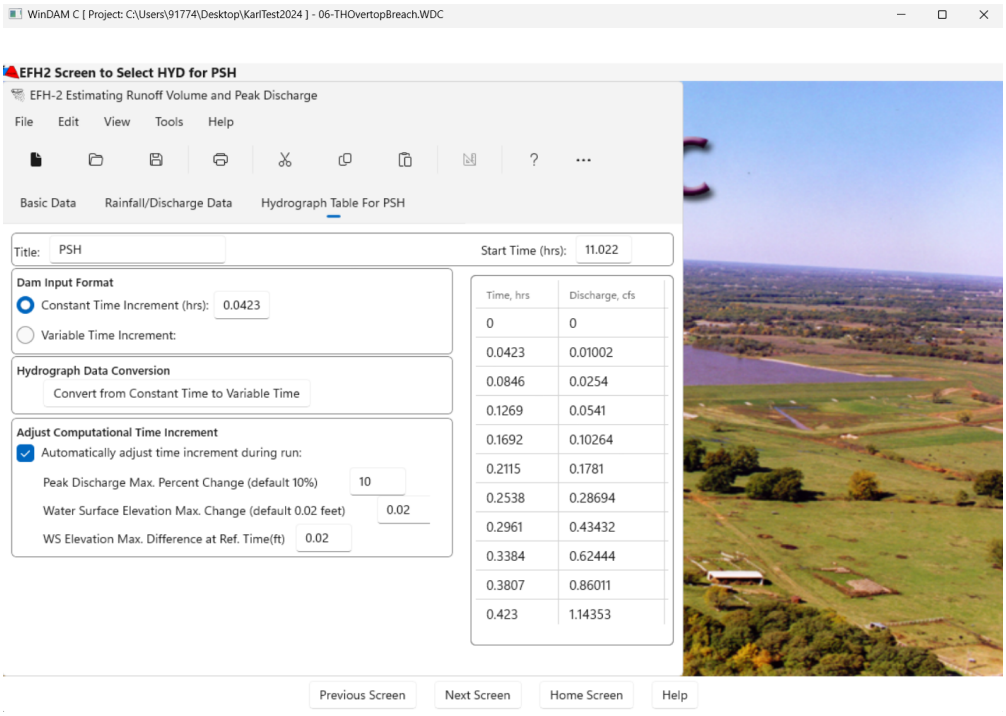


Figure B.1: *EFH-2 hydrograph screen for the PSH, with the automatic time-increment adjustment and its three limits (Section 3.7.1).*

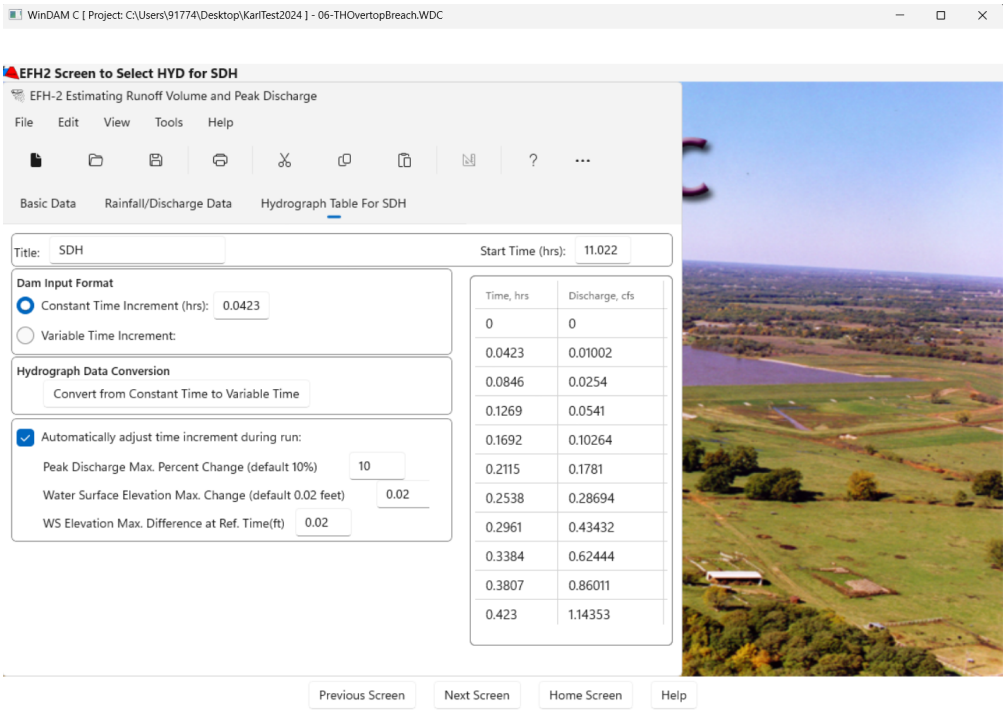


Figure B.2: *EFH-2 hydrograph screen for the SDH, with the automatic time-increment adjustment and its three limits (Section 3.7.1).*

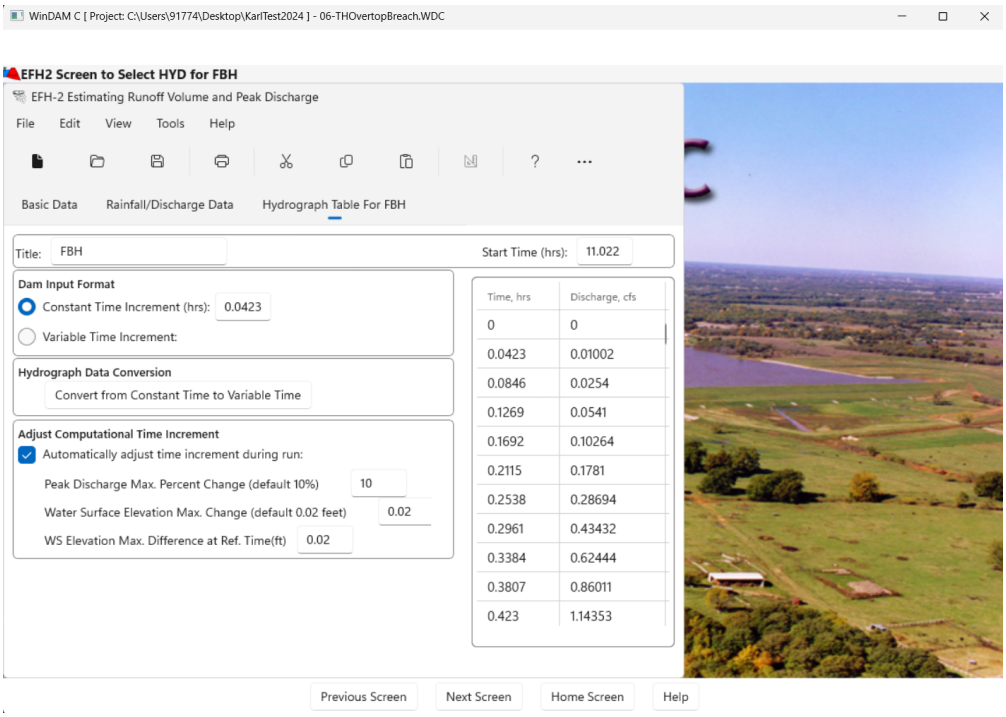


Figure B.3: *EFH-2 hydrograph screen for the FBH, with the automatic time-increment adjustment and its three limits (Section 3.7.1).*

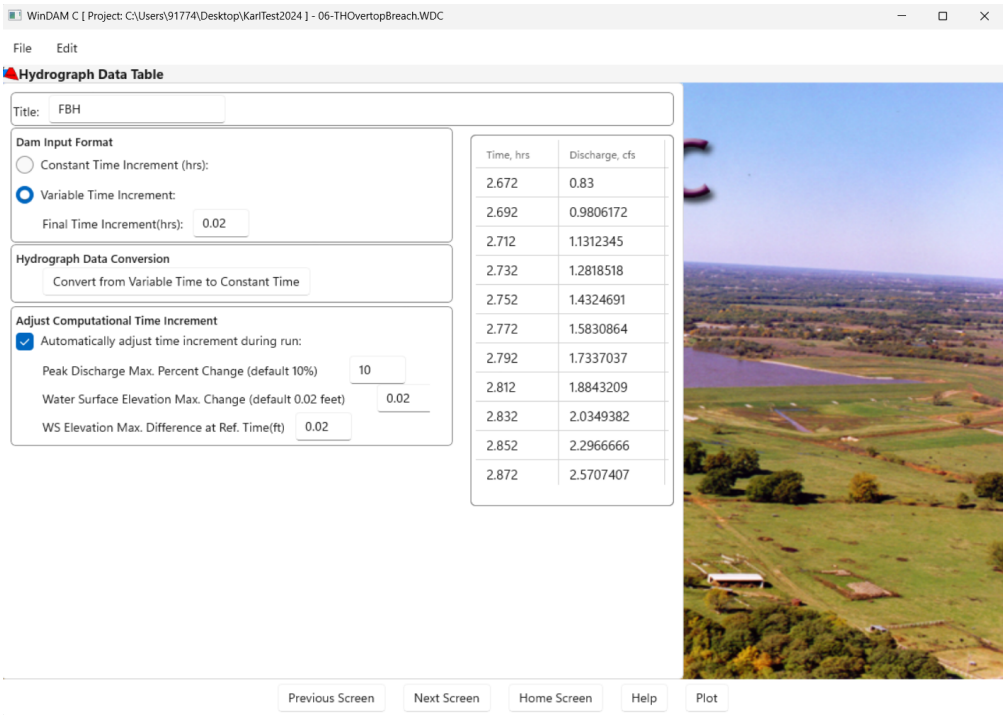


Figure B.4: *Main hydrograph screen, with the conversion from a variable to a constant time increment (Section 3.7.2).*

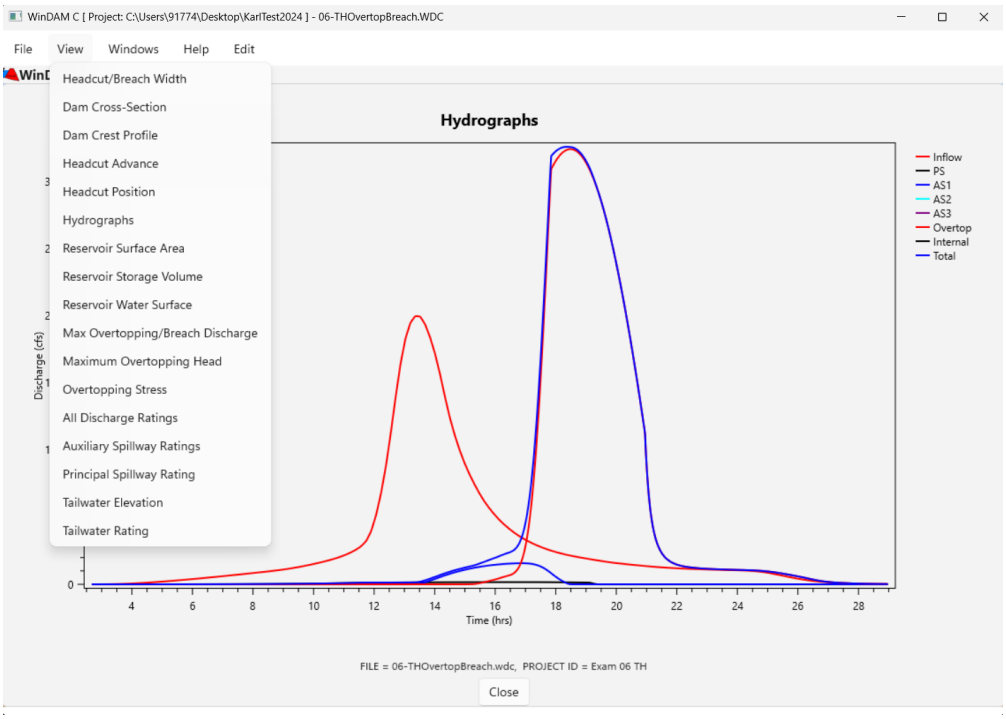


Figure B.5: *View menu listing the results graphs, over the Hydrographs graph of a run (Section 3.7.3).*

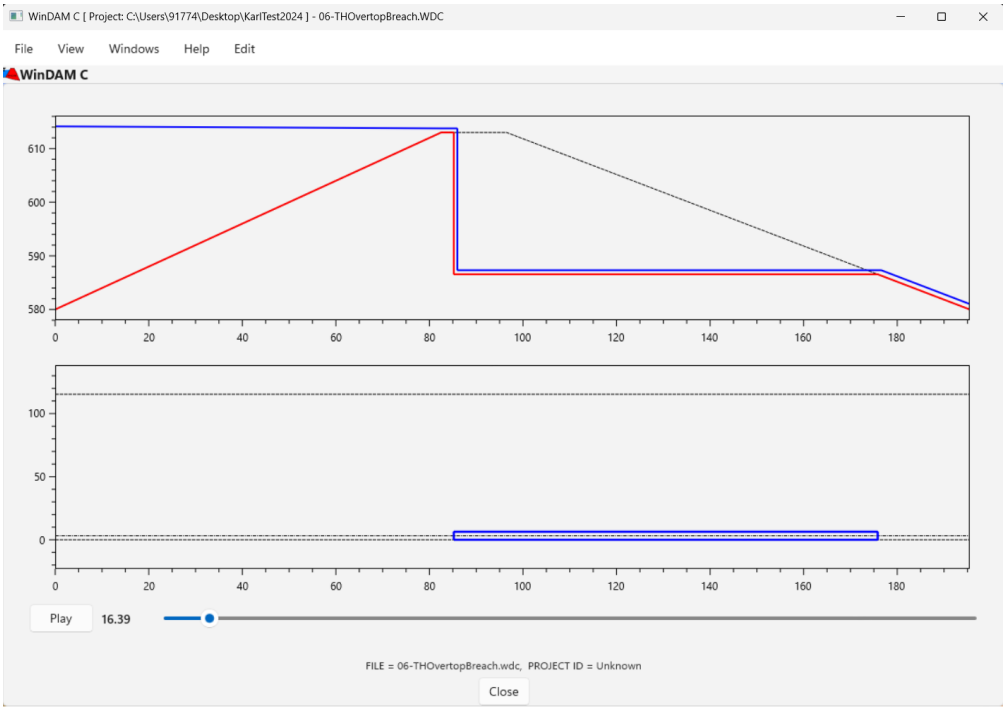


Figure B.6: *Headcut Advance graph: side view of the dam (top), plan view of the breach (bottom), and the Play button and time slider (Section 3.7.4).*

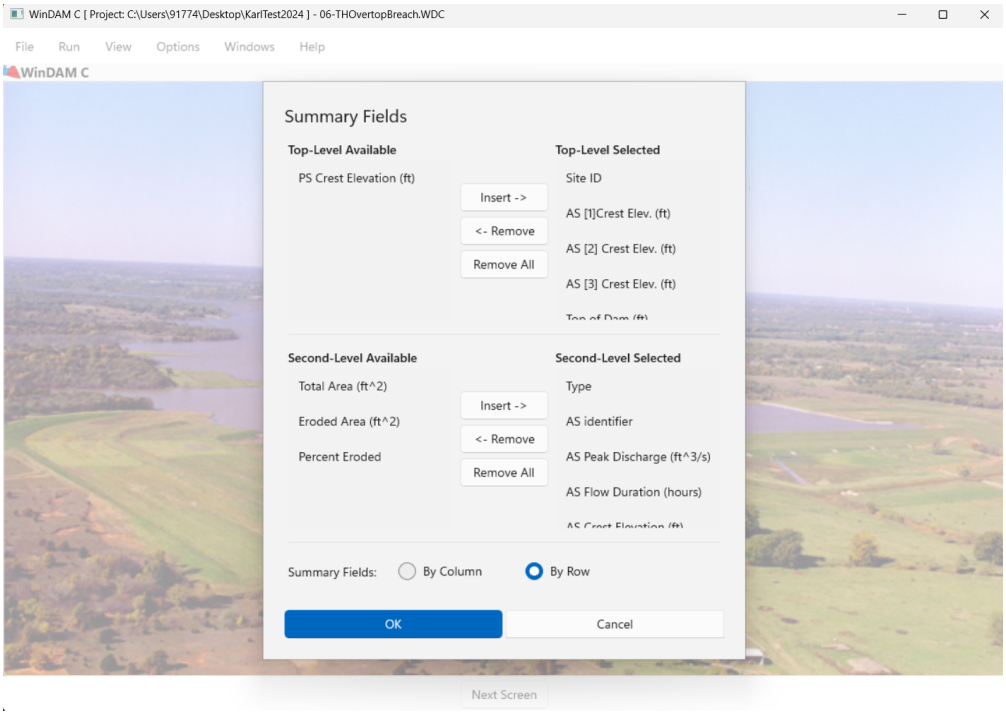


Figure B.7: *Summary Fields dialog for choosing the fields of the dam-level and auxiliary-spillway summary tables (Section 3.7.5).*

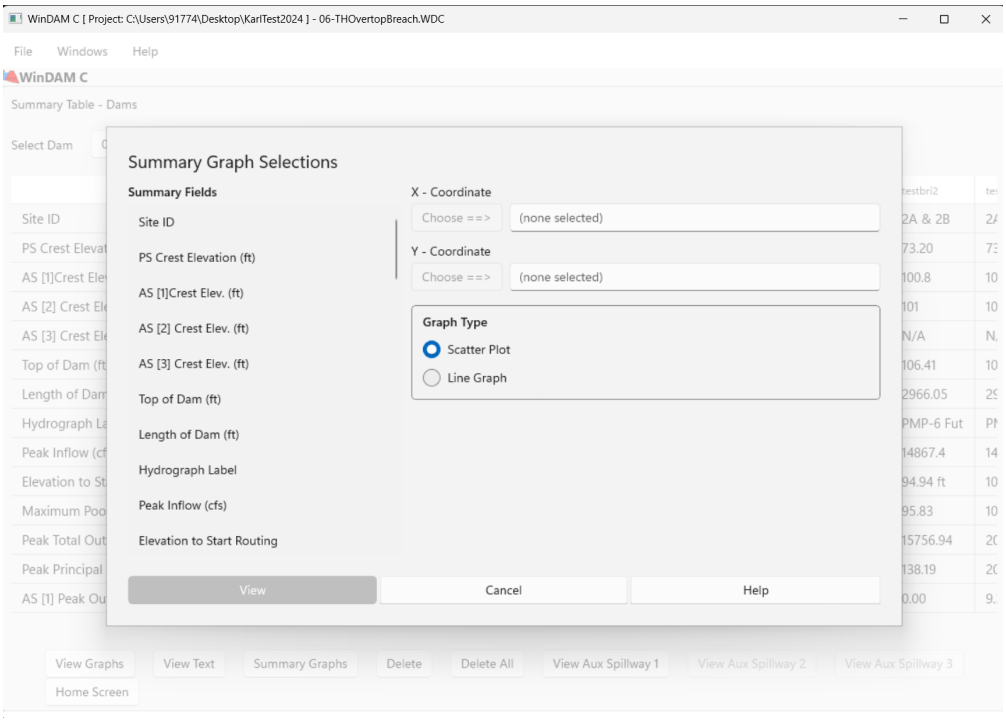


Figure B.8: *Summary Graph Selections dialog for choosing the two fields and the graph type (Section 3.7.6).*

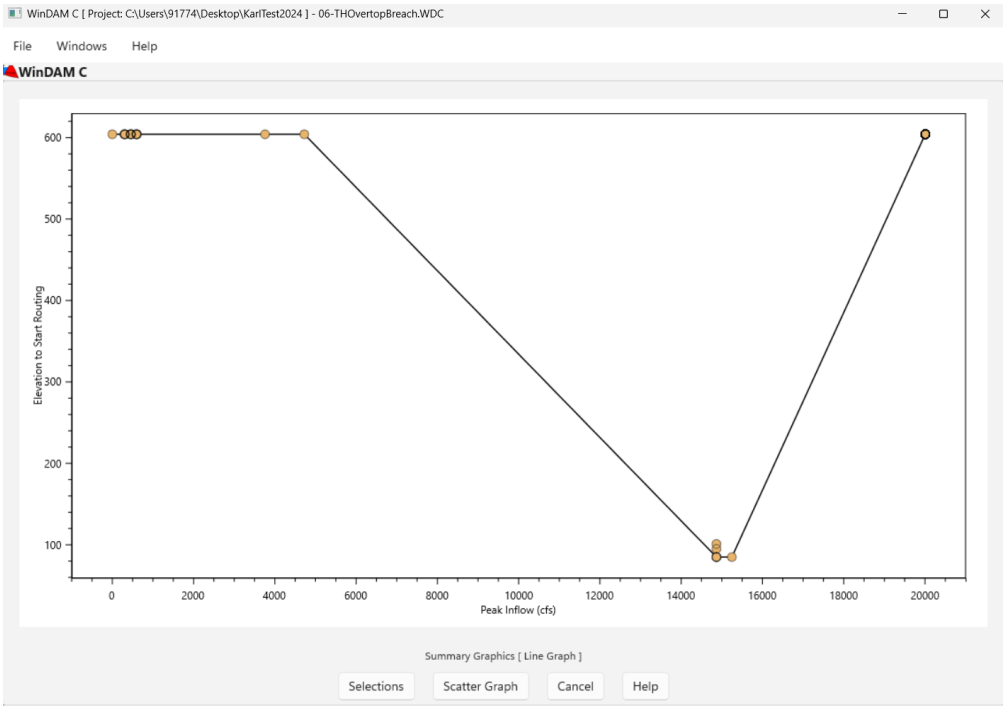


Figure B.9: *Summary graph across runs, drawn as a line graph (Section 3.7.6).*

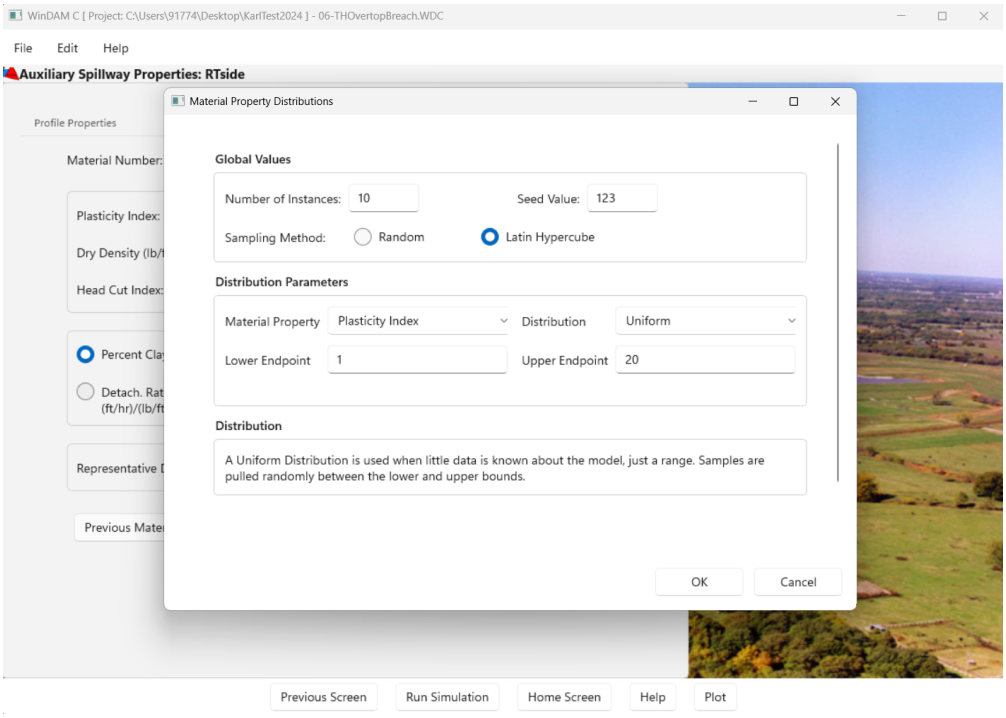


Figure B.10: *Material Property Distributions dialog with the sampling settings of an auxiliary spillway (Section 3.7.7).*