403 Poyntz Ave., Suite B
Manhattan, KS 66502, USA
Phone: +1-785-770-8511
Email: support@thunderheadeng.com
Web: https://www.thunderheadeng.com

Pathfinder Software Quality Assurance Plan

Version: 2020-4
Last Modified:

Table of Contents

Preface

The purpose of this document is to describe the policies and procedures for developing and maintaining Pathfinder. Such a document is commonly referred to as a *Software Quality Assurance Plan*, known through the rest of this document as the "**SQAP**". This document will be updated as the necessity arises and will establish and provide the basis for a uniform and concise standard of practice for Pathfinder development.

Download the PDF Version of this page.

This plan is based in part on IEEE Standard 828 ("IEEE 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering" 2012) and DOE Order 414.1D (Palay 2013). The Pathfinder Technical Reference Guide is based in part on IEEE Standard 26511 ("IEEE 26511-2018 - Systems and Software Engineering — Requirements for Managers of Information for Users of Systems, Software, and Services" 2018). The Pathfinder Verification and Validation (V&V) Guide is based in part on IEEE Standard 1012 ("IEEE 1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation" 2016) and ISO 16730-1 ("ISO 16730-1:2015E - Fire Safety Engineering — Procedures and Requirements for Verification and Validation of Calculation Methods" 2015).

As progress is made in understanding human psychology, pedestrian movement and crowd modeling, the model needs to be updated and improved, but still shown to reliably predict the phenomena for which it was originally designed.

The process to ensure this reliability includes the following elements:

- Relevant documentation for the model:
  - *Pathfinder Technical Reference* (*Pathfinder Technical Reference*, n.d.) describes the technical details, mathematical model and numerical methods for the Pathfinder model.
  - *Pathfinder Verification and Validation* (*Pathfinder Verification and Validation*, n.d.) documents Pathfinder verification and validation work, respectively.
  - *Pathfinder User Manual* (*Pathfinder User Manual*, n.d.) provides instructions for using the simulation preprocessor (primary user interface) for Pathfinder.
  - *Pathfinder Results User Manual* (*Pathfinder Results User Manual*, n.d.) provides instructions for using the postprocessing visualization software included with Pathfinder.
- Description of the development process for the model.
- Control and tracking of the source code using a secure centralized project repository.
- An automated build, verification, validation, and regression testing process then helps the Pathfinder development team by performing automatic error checking and collecting

performance and accuracy statistics between Pathfinder revisions.

- Details of the review process used by the developers to ensure the quality of the model and related publications.

This document applies only to the program Pathfinder and related Thunderhead Engineering publications and Internet sites.

Disclaimer

Thunderhead Engineering makes no warranty, expressed or implied, to users of this software, and accepts no responsibility for its use. Users of this software assume sole responsibility under Federal law for determining the appropriateness of its use in any particular application; for any conclusions drawn from the results of its use; and for any actions taken or not taken as a result of analyses performed using these tools. This software is intended only to supplement the informed judgment of the qualified user. The software package is a computer model that may or may not have predictive capability when applied to a specific set of factual circumstances. Lack of accurate predictions by the model could lead to erroneous conclusions. All results should be evaluated by an informed user.

Trademarks

All other product or company names that are mentioned in this publication are tradenames, trademarks, or registered trademarks of their respective owners.

Chapter 1. Pathfinder Overview

The main purpose of this software quality assurance plan (SQAP) document is to describe the process by which the model software is developed and maintained. As progress is made in understanding human behavior and locomotion, Pathfinder needs to be updated and improved, but still shown to reliably predict the phenomena for which it was originally designed. This document describes the processes used during the development and deployment of Pathfinder and associated publications.

This document is intended to guide planning for modifications to the model, provide required reviews for both software and associated documentation of the model, define testing to be conducted prior to the release of an updated model, describe problem reporting and resolution procedures, and ensure all records, source code, and released software is kept available for the life of the code. The Institute of Electrical and Electronics Engineers (IEEE) standards 730 (“IEEE 730-2014 - IEEE Standard for Software Quality Assurance Processes” 2014) and 828 (“IEEE 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering” 2012) define best practices in important areas of software quality assurance, and provides a foundation for the structure and content of this plan.

This section provides a brief overview of Pathfinder, including its history, developers, features, and applications.

1.1. Model Type

Pathfinder is an agent-based egress simulator that uses steering behaviors to model occupant motion. It consists of three components: preprocessor graphical user interface (GUI), simulator, and Results postprocessor. The GUI and simulator are bundled together into what is commonly referred to as "Pathfinder". Pathfinder Results is included as part of the Pathfinder installation process, but can be opened and operated independently from the Pathfinder GUI.

Pathfinder provides two primary options for occupant motion: an SFPE mode and a steering mode.

- **SFPE Mode** implements the concepts in the SFPE Handbook of Fire Protection Engineering [Nelson and Mowrer, 2002]. This is a flow model, where walking speeds are determined by occupant density within each room and flow through doors is controlled by door width.
- **Steering Mode** is based on the idea of inverse steering behaviors. Pathfinder’s steering mode allows more complex behavior to naturally emerge as a byproduct of the movement algorithms - eliminating the need for explicit door queues and density calculations.

For more detailed information see the Pathfinder Technical Reference.

1.2. Intended Uses

Throughout its development, Pathfinder has been aimed at simulating human movement within an indoor or outdoor environment. Although Pathfinder was originally designed for evacuation simulations, it can now be used for some general pedestrian movement simulations that do not necessarily include evacuation.

Pathfinder can be used to model the following phenomena and more:

- Specify agent characteristics through profiles
- Goal seeking behavior based on agent behaviors
- Human movement on user identified navigation surfaces.
- Movement on stairs, ramps, escalators and moving walkways
- Shelter in place with identified refuge areas
- Grouping behavior
- Use of elevators as a component of evacuation strategy
- Assisted evacuation
- Incoming agent flow from room, door or specified area sources
- Calculation of Fractional Effective Dose (FED) for agents moving in hazardous atmosphere
- Door choice based on queue size and door flow rate
- Time based availability of doors and movement speeds on surfaces

1.3. Input Parameters

Input parameters to describe a particular model are conveyed to the simulator via a single text file created by the user through the Pathfinder GUI. The input file specification is detailed in the *Pathfinder Input File Format* section of the Pathfinder Technical Reference document. A complete description of the input parameters specified by Pathfinder can be found throughout the Pathfinder User Manual.

1.4. Pathfinder Output

The Pathfinder simulator computes the movement of occupants over time and some associated quantities related to their path over the navigation surface.

Some output files record time histories of various quantities and are saved in simple comma-delimited (CSV) text files. Data within these files can be plotted using a spreadsheet program or through the 2D plotting tool built into Pathfinder.

More complex information related to the movement of occupants and associated surface contour data are visualized with the Pathfinder Results program. Pathfinder Results is a postprocessing tool specifically designed to analyze 3D Results data generated by Pathfinder and included with the Pathfinder installation.

A complete list of Pathfinder output quantities and formats is given in the *Results* section of the Pathfinder User Manual. Details on the visualization and postprocessing tool are found in the Pathfinder Results User Manual.

1.5. Governing Equations, Assumptions and Numerics

Detailed information regarding the assumptions and governing equations associated with the model is provided in the Pathfinder Technical Reference.

There is a fundamental assumption that guides development of Pathfinder, which is if there is a bias in the results of Pathfinder, that it should fall on the side of conservatism.

1.6. Limitations

Although Pathfinder can address most evacuation and pedestrian movement scenarios, there are limitations in all of its various algorithms. The Pathfinder Technical Reference combined with the Pathfinder Verification and Validation document are intended to allow the user to make an informed decision about the suitability of Pathfinder to their specific application. Thunderhead technical support staff are also available to respond to questions or to provide further details as needed.

Chapter 2. Configuration Management

2.1. Project Management

Pathfinder is developed by Thunderhead Engineering, Headquartered in Manhattan, Kansas, USA. Within the organization there are individuals and groups who coordinate on various operations related to planning, design, development, documentation, testing, verification, validation, release processes, training and technical support.

Thunderhead staff share in the development and maintenance responsibilities, which include:

- Developing new algorithms and improving program functionality
- Answering user questions
- Responding to bug reports
- Issuing periodic updates to the officially released versions of the programs
- Maintaining the Technical Reference and User Guides
- Maintaining a suite of sample cases to demonstrate model use
- Maintaining a suite of verification and validation cases to test model accuracy and reliability
- Creating and maintaining content related to Pathfinder on the Thunderhead website.

Decisions concerning various tasks are made individually or by consensus of the governing board and/or staff members. In the event of a disagreement over a technical case, the resolution is facilitated by the corporate board. Non-technical cases are addressed by the president of the company or by the identified staff member authorized to make the decision.

Review and approval of software and documentation is part of the standard testing, review and release process for any report or other product developed by Thunderhead. Content of the website is the responsibility of the staff member assigned to manage it.

Pathfinder is distributed through the Thunderhead website and through authorized distributors identified on the website.

2.2. Document Identification and Control

Document identification and control consists of placing all project files in a central location and maintaining a record of changes to those files, the central location is known as the *Repository*.

2.2.1. Project Repository

A Perforce HelixCore server provides the Repository for Pathfinder source code, documentation and related assets. This centralized repository containing all project files resides on an internal server located at the main office in Kansas.

In Perforce, the code changes are organized into changelists. The Perforce application shows the depot, the workspace, pending changelists, and the history of submitted changelists.

Each authorized staff member has an account and password access to the Thunderhead repository. As a 'closed source' project only authorized staff members can connect and commit changes to the repository.

Files stored in this Repository are backed up on site and a redundant backup is synchronized with a cloud storage location for disaster recovery.

2.2.2. Version Identification and Numbering

Official versions of Pathfinder released by Thunderhead are identified with specific version numbers tied to year, major version, month and date in the repository and are identified on the download page. Only the latest release of Pathfinder is made available on the download page, but earlier versions can be downloaded upon request.

Pathfinder was publicly released for the first time on April 17th, 2009 as version `2009.1.0417`. Each release is marked by the year, major version, month and date of release. For example, the version shown above (2009.1.0417) provides the year 2009, major release number 1 of the year, the 4th month (April) and the 17th day of the month. Minor maintenance releases after a major release would be identified by a change to the last four digits (MMDD) only. There is a general policy to not release a maintenance release on the same day as a previous release, usually this delay is a natural byproduct of the time required for development, testing and release process.

Details of changes made for each release is located on the Pathfinder Release Notes page of the Thunderhead website.

Pathfinder documentation is identified by year and version on the title page of each document. Release notes indicate whether the changes affect a particular model feature.

The Pathfinder manuals are stored in a repository, these files are exported to HTML and PDF format (the PDF file is available through a link at the top of the HTML version), and versions prior to the more advanced online documentation system are available through the V&V Archive page of the Thunderhead support website.

Thunderhead personnel edit the manuals as a normal part of development. Different editions of the manuals are distinguished by year and version number.

Chapter 3. Software Requirements Identification, Design, Implementation, and Testing

Changes are made to the Pathfinder source code on a regular basis, and tracked via revision control software. However, these incremental changes do not constitute a change to the version number. After the developers determine that adequate changes have been made to the source, they release a new version, `2020.1.0101` to `2020.3.0202`, for example. This might happen only a few times a year, when significant improvements have been made to the model usability, physics or numerics.

The software development process for Pathfinder can be defined in six basic steps:

1. Identify and document the need for a change, then enter the request into the FogBugz tracking system.
2. Review and assign a status and priority to each change request.
3. Implement the change.
4. Before accepting any change into the repository, the code changes are reviewed and final approval is given by the change request initiator.
5. Nightly testing using a suite of test calculations to ensure that any changes result in valid results.
6. A new version of released after a formal review of technical documentation and software.

3.1. Creating a Change Request

A change request is initiated by opening a new case in the FogBugz case tracker. The case typically contains information about the issues to be resolved or the features to be added. If the change request was created based on communication from an external user, then a reference to that inquiry case is added to the case information. Files, images or other supporting documentation can be attached to the case. The new case will be given a status of **Active (Needs Review)** until it is reviewed during a weekly triage meeting.

3.2. Review and Assign Priority

During the New Bug Triage session, the case status will be assigned. This includes an estimate of time required for the work to be done, tags to help sort cases by theme, and a priority for implementation. Any bug that would lead to an error in the simulation is given a critical priority and is resolved in the next release.

3.3. Implement Changes

Once a developer has addressed a change request, the modified files are committed to the Repository with a short description of the change, ideally containing the case number and name. For example, the description could be `[Pathfinder: inferno] Fixed case 12345 "Broken simulator"`. In this example, there would be a bug case in the tracking system with number 12345 and title of "Broken simulator". The description would typically also contain a short description of how the bug was fixed or what change was made.

3.4. Reviews of Software Change

Prior to final implementation of a new feature or bug fix, a review of the proposed modification and documentation is conducted. Other members of the development team perform a code review. Final approval is given by the originator of the FogBugz case who closes the case in the system as "Implemented".

3.5. Automated Software Quality Testing

The Pathfinder source codes undergo an automated build, verification, validation, and regression testing process, which is a continuous integration system that automates the build/testing cycle for a project. This automated system is referred to as TeamCity. The TeamCity build/test process helps the Pathfinder development team by performing automatic error checking and collecting performance statistics between Pathfinder revisions.

The automated built/test verification process runs on a regular schedule (nightly) to ensure that Pathfinder are free of compiler errors, verification errors, or any other errors that would result in a failure during the build/test cycle. For future reference, the results of the nightly verification process are archived and tagged with the appropriate revision number that was used.

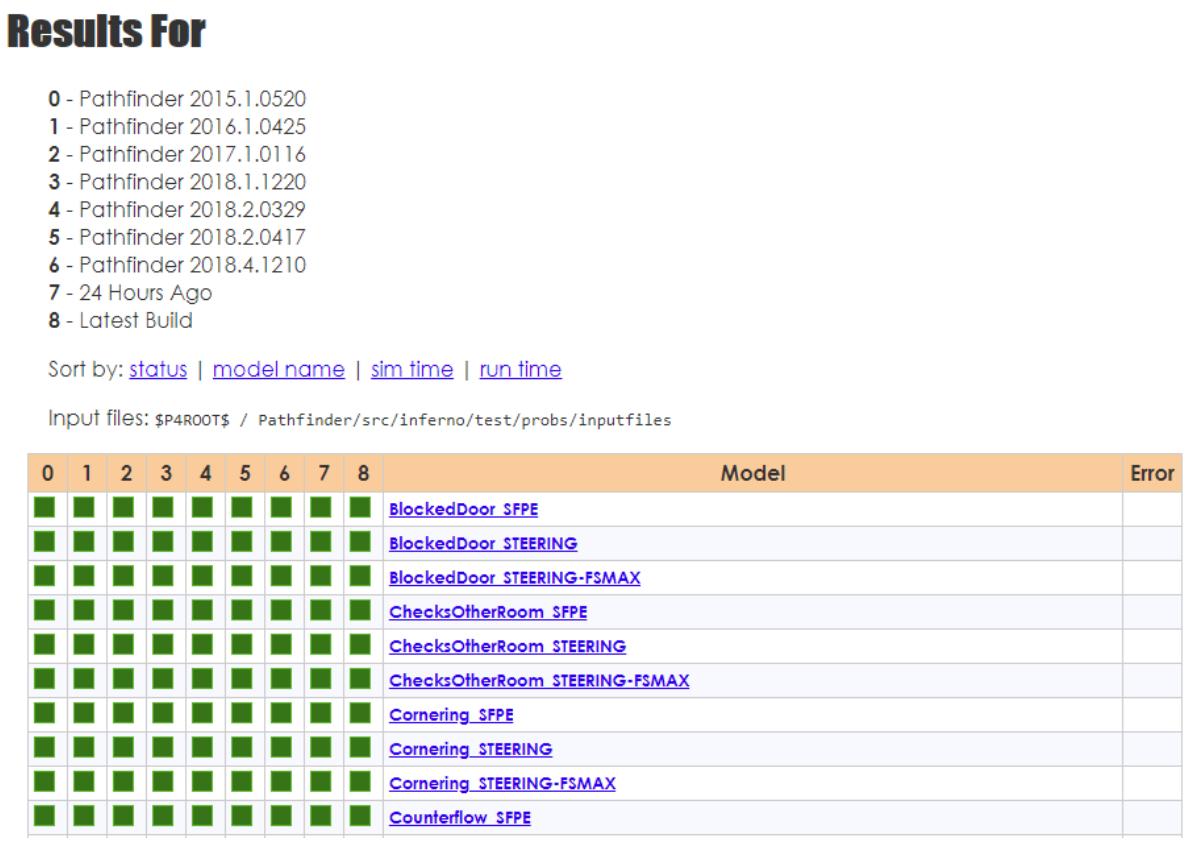

Figure 1. Test Case Results Grid

Upon completion of the automated build/test process, the results of the build/test process (see example in Figure 1), are dispatched to the development team. In the event of a failure, the build/test process continues when possible, and the appropriate error logs are dispatched to the development team. The automated build/test process consists of three build stages.

3.5.1. Stage 1: Checkout of Pathfinder codebase

In the first stage, the newest version of the Pathfinder source code is retrieved from the Pathfinder repository. At this stage, the TeamCity script ensures that all temporary files are removed and that a clean copy of the repository is used for the later stages, as if an end user were downloading the repository for the first time.

3.5.2. Stage 2: Compilation of Pathfinder

Pathfinder is compiled and any compiler warnings are logged at this time, and the build/test process continues after no major compilation errors are found.

3.5.3. Stage 3: Run unit test suite

All of the cases in the unit test suite are invoked using Pathfinder compiled in Stage 2. After all of

the cases run to completion, the unit testing output is compiled into a report for review. If errors are discovered in this process, email and instant message alerts are sent to the development group. If it passes all tests, then subsequent processes are allowed to continue.

3.6. Issuing New Releases

The decision to change versions of the software is made by consensus of the development team, usually after it is determined that enough changes have been made to warrant a new release. Once the decision is made, the new version is given a number, it is tested, and then posted to the official download page.

3.6.1. Testing a New Release

Each proposed release undergoes testing, to pass a suite of verification and validation tests. The verification suite consists of a range of calculations, from ones that just test a feature to ones that compare Pathfinder results with analytical solutions of the governing equations. The validation suite consists of simulations of a wide range of experiments.

A major release occurs every three to six months. Each major release is tested with both the verification and validation suites. The results are plotted and compared to the experimental measurements using an automated plotting package that redraws all of the plots and graphs in the Pathfinder Verification and Validation document.

A maintenance release occurs as needed. Each maintenance release is tested with the verification suite during the nightly TeamCity testing. Maintenance releases are intended for quick bug fixes, documentation clarification, and usability cases. There should be no changes in code functionality with a maintenance release, although the bug fixes might change results slightly.

3.6.2. Announcing a New Version

Following successful completion of the required baseline testing, a new version is released. Prior to release, the version identification information within the Pathfinder source code is updated. Pathfinder documentation is updated to include the new version number. The new version is compiled and new installation packages are uploaded to the Pathfinder download page. Prior versions are archived and deprecated.

Chapter 4. Software Risk Management

The primary risk management tool for software developed and released by Thunderhead is the review process for software, documents, and other products of Thunderhead outlined above. Official approval is required prior to release of the model for general use. Additional processes to minimize risk are identified below.

4.1. Media Control

Release versions of Pathfinder in the English Language are available on the Pathfinder website which is also under revision control. Release versions of Pathfinder that have been built with translated interface strings are available through the distributor who manages the translation.

As part of its model development, Thunderhead maintains an internal system for version control and history of both the Pathfinder source code and of pre-release and release packages for the software.

These computer systems are available only to specified personnel.

4.2. Supplier Control

Pathfinder includes third-party software libraries. These libraries are tested in context of the necessary operations within Pathfinder.

4.3. Records Collection, Maintenance, and Retention

In addition to the identified configuration items included in the project repositories, all software, documentation, and Software Quality Assurance documents are retained, typically in electronic form.

Software and documentation is also maintained and archived on the Thunderhead support website as partial archive of older assets.

Thunderhead management approval is required prior to destruction of old or out-of-date records.

4.4. Training

No specific training is identified for use of this plan. Details of training requirements for use of the model included in the Pathfinder User Guide is applicable to developers of the model as well.

Bibliography

Pathfinder Technical Reference. n.d. 403 Poyntz Avenue, Suite B, Manhattan, KS 66502, USA: Thunderhead Engineering. https://support.thunderheadeng.com/docs/pathfinder/latest/technical-reference-manual/.

Pathfinder Verification and Validation. n.d. 403 Poyntz Avenue, Suite B, Manhattan, KS 66502, USA: Thunderhead Engineering. https://support.thunderheadeng.com/docs/pathfinder/latest/verification-validation/.

Pathfinder User Manual. n.d. 403 Poyntz Avenue, Suite B, Manhattan, KS 66502, USA: Thunderhead Engineering. https://support.thunderheadeng.com/docs/pathfinder/latest/user-manual/.

Pathfinder Results User Manual. n.d. 403 Poyntz Avenue, Suite B, Manhattan, KS 66502, USA: Thunderhead Engineering. https://support.thunderheadeng.com/docs/pathfinder/latest/results-user-manual/.

“IEEE 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering.” 2012. Institute of Electrical and Electronics Engineers.

“IEEE 730-2014 - IEEE Standard for Software Quality Assurance Processes.” 2014. Institute of Electrical and Electronics Engineers.

“ISO 16730-1:2015(E) - Fire Safety Engineering — Procedures and Requirements for Verification and Validation of Calculation Methods.” 2015. International Organization for Standardization.

“IEEE 1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation.” 2016. Institute of Electrical and Electronics Engineers.

“IEEE 26511-2018 - Systems and Software Engineering — Requirements for Managers of Information for Users of Systems, Software, and Services.” 2018. Institute of Electrical and Electronics Engineers.

Palay, Christian. 2013. “Quality Assurance.” Order. Washington, D.C.: U.S. Department of Energy. http://bit.ly/36GcpQc.