Contents

Introduction:

The NETWORK PATH ANALYZER is a program designed to allow users to enter various project activities along with their durations and dependencies (defined as activities which must occur prior to another activity) into the program which will then calculate the various work paths that could possibly flow through the project – particularly identifying the critical or longest path.

This User Guide will explain the topics outlined in the Table-of-Contents including: system requirements, installation, how to start the program and input data including any limitations on the forms of the entered data and what an error might look like if the data is entered incorrectly, how to run the program either without making a report of the results or making a report of the results including how to alter already-entered data and rerun the program and how to delete entered data and end the program. Examples runs of the program are provided and various error messages a user might encounter are identified and explained.

Overview:

To use the program a user inputs their data (activities/durations/dependencies) in a presented Entry Table. An unlimited number of activities is allowed. If any data is entered incorrectly an error message is shown and re-entry is requested. After running the program, output is either presented in table format showing all various activity paths through the project, or is provided to the user by way of a report in the form of a .txt file with the critical path being the first pathway shown in either case. If the project as entered allows for a “cycle” or looping back upon itself, an error message will be shown as cycles are not permitted.

System Requirements and Installation:

Requirements for running the program:

In order to run the NETWORK PATH ANALYZER program a user needs to have the following installed on their computer:

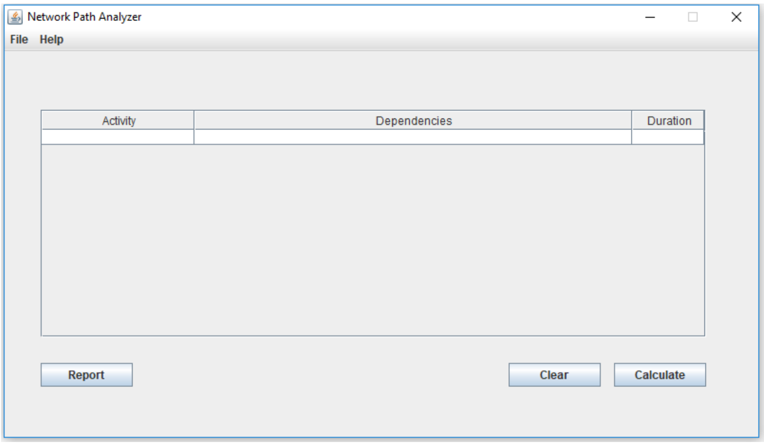
- an active JAVA RUNTIME ENVIRONMENT, and
- a reader that has the capability of displaying .pdf files (both of these are present in most computers), and
- the NETWORK PATH ANALYZER program’s zipped folder containing the program’s NetworkPathAnalyzer.jar file and two .pdf files.

Installation:

The user will be given a zipped folder containing the materials for the NETWORK PATH ANALYZER. To run the program, the user unzips the zipped folder and merely double-clicks the NetworkPathAnalyzer.jar file inside the folder in order to initiate the program.

Getting Started:

Upon opening the NetworkPathAnalyzer.jar file the user will be presented with an Entry Table:



Input:

Once presented with the Entry Table, the user will enter the various activities comprising the project into the appropriate cell of the Entry Table:

- activity names can include characters, number, or spaces or be totally numerical;
- duration of the activity as an integer (example: 1, 5, 13, 22) (*see* example error below);
- list of dependencies (predecessors). In other words, what activity or activities must be completed prior to activity. Note that a listed dependency must be entered exactly as the activity was initially entered or an error will be shown (*see* example error below).
- Each of these data entries can be changed before or after the program is run.

Data Entry Limitations:

Note that all activities must be connected to another activity. In other words each activity must either serve as a predecessor or have a predecessor. The starting activity (there may be more than one starting activity) must not depend on any prior activity, and the final activity must not serve as a predecessor to any activity. Activities do not need to be entered in any particular order.

Maximum Number of Activities:

There is no maximum number of activities and/or dependencies(predecessors).

Error Detection:

If an error is detected anywhere in the input, the user will be notified and re-prompted for new input (*see* example error below).

Running the Program with or without a Written Report, Ending the Program:

Running the Program:

Once the user has completed their input, there are two different ways to run the program:

- click on the [Calculate] button on the lower right of the Entry Table frame to begin processing and generate an on-screen table of results; or
- click on the [Report] button on the lower left of the Entry Table frame after which the program will perform the calculation and provide the user with a written report of the results in the form of a text file. The user provides a name for the report.

Re-Running the Program without Exiting Completely:

After the program has run the user has two options for re-running the Program:

- If the user ran the program by clicking on the [Calculate] button, to rerun they can click the [Close] button on the lower right of the Results Table frame to close the results frame and return to the Entry Table;
- If the user ran the program by Clicking the [Report] button, to rerun the program if the user decides not to run a report they can click the [Cancel] button on the lower right of the report pop-up and they will be returned to the Entry Table, or if they do decide to run a report once the report is generated the program will automatically take the user back to the Entry Table upon which the user has multiple options:
    - once either of the above is done, the user can retain and/or modify the data currently entered in the Entry Table and modify it as needed by double-clicking inside the cell to be edited; or
    - the user can delete the currently-entered data by clicking the [Clear] button data and restart the data entry process.

Ending the Program and Exiting Completely:

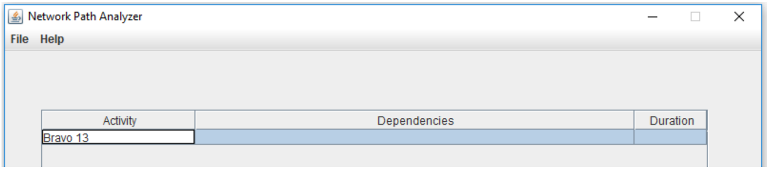
Once the user has completed their input and run the program either a Results Table or Report will be generated. To end the Program:

- if the user ran the program by clicking the [Calculate] button: click the [Close] button on the Results Frame to close the frame; then, however the user ran the program
- click the [X] in the upper-right corner of the entry frame to end the program.
- to restart the program after exiting, double-click the NetworkPathAnalyzer.jar file.

Example Run of the Program Number 1:

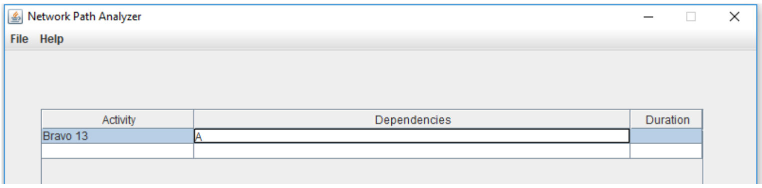
Entering Path Information:

Start by entering the name of an activity (the user does not need to start with the first activity of the project and activities do not need to be entered in any particular order) into the Entry Table. Activity names can have letters, numbers, characters, and spaces (example allowable activity names: A, Bravo 13, Charlie, D23, Foxtrot-5, Golf_19, 625).



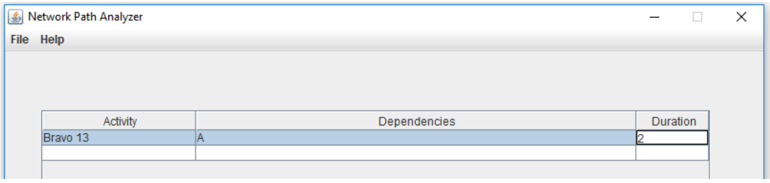
Activity	Dependencies	Duration
Bravo 13		

Next, enter any dependencies this activity will have. If more than one dependency, separate entries using commas. If no dependencies are entered for a particular activity, then that activity is assumed to be the start activity. Note that dependencies must match their associated activity names exactly. If they do not, an error message indicating there is not such activity name will be shown (see example error below), and the user will need to correct the entry.



Activity	Dependencies	Duration
Bravo 13	A	

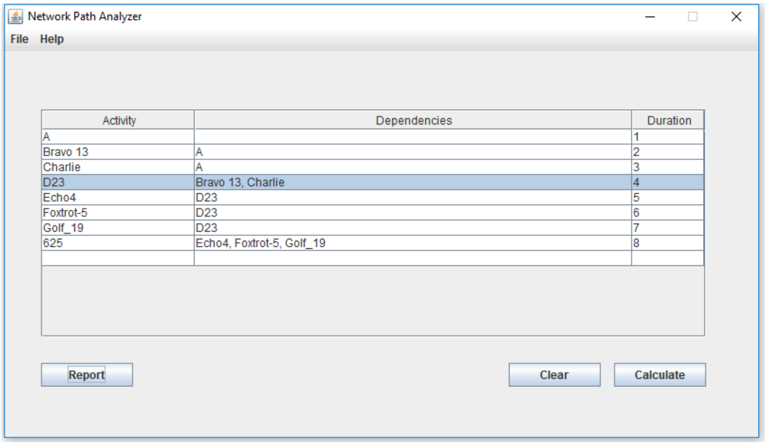
Finally, enter the duration value rounded up to the nearest integer.



Activity	Dependencies	Duration
Bravo 13	A	2

Calculating Paths/Output:

After entering all activities with their corresponding dependencies and durations the user is now ready to calculate all available paths through the project. For an example, let us assume the user entered the following data (note the different allowable forms of activity name):

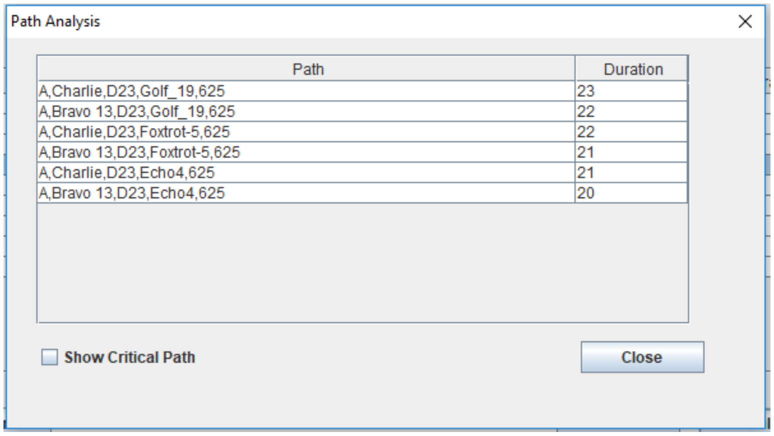


Activity	Dependencies	Duration
A		1
Bravo 13	A	2
Charlie	A	3
D23	Bravo 13, Charlie	4
Echo4	D23	5
Foxtrot-5	D23	6
Golf_19	D23	7
625	Echo4, Foxtrot-5, Golf_19	8

Once data is properly entered, there are two ways to have the Program calculate the various paths through the user's project:

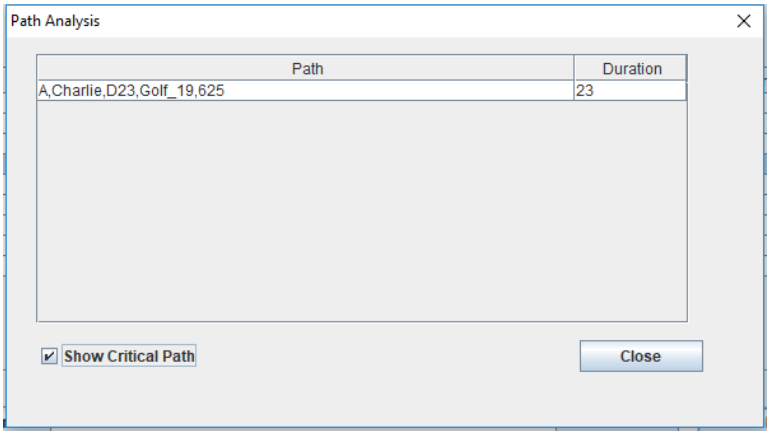
1) Click the [Calculate] button at the bottom-right of the Entry Table frame

The Program will output the following: a list of all possible activity paths through the project, with the Critical Path being listed first. Paths after the critical path are listed in descending order by duration. The critical path will be the first path displayed on the list. The data in this example is fine, but if the user enters an unallowable project, for example one with a cycle, they will see an error message (*see* example error below).



Path	Duration
A,Charlie,D23,Golf_19,625	23
A,Bravo 13,D23,Golf_19,625	22
A,Charlie,D23,Foxtrot-5,625	22
A,Bravo 13,D23,Foxtrot-5,625	21
A,Charlie,D23,Echo4,625	21
A,Bravo 13,D23,Echo4,625	20

If the user wants the program to show the Critical Path only, in the results table click the [Show Critical Path] check box and all results but the critical path will be masked. To re-list all paths, simply uncheck the Show Critical Path box and the other paths will reappear.

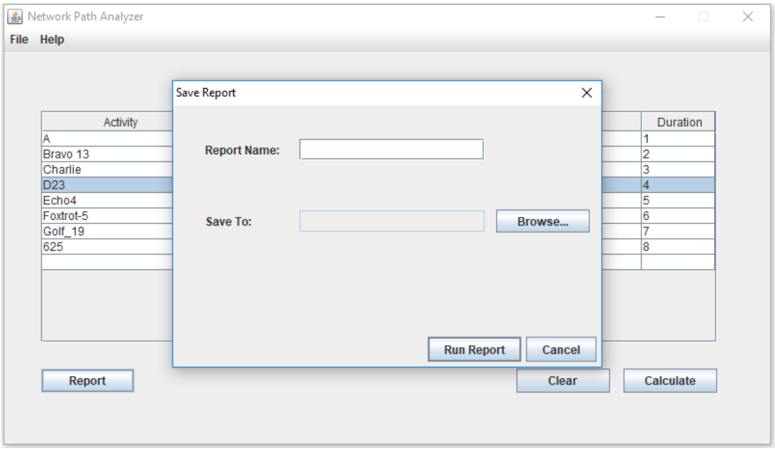


Path	Duration
A,Charlie,D23,Golf_19,625	23

☑ Show Critical Path

2) Click the [Report] button at the bottom-left of the Entry Table frame

An alternative way to run the Program also allows the user to save the entered data and result of the program's analysis by creating a report. The report will be a text file that the user can name. The report will appear as follows, including all of the data the user entered and all calculated paths through the project. Upon clicking the [Report] button on the lower left of the Entry Table frame, the user will be presented with the following report pop-up asking the user to provide a name for the report and to identify where on the user's computer the report is to be saved:



Activity	Duration
A	1
Bravo 13	2
Charlie	3
D23	4
Echo4	5
Foxtrot-5	6
Golf_19	7
625	8

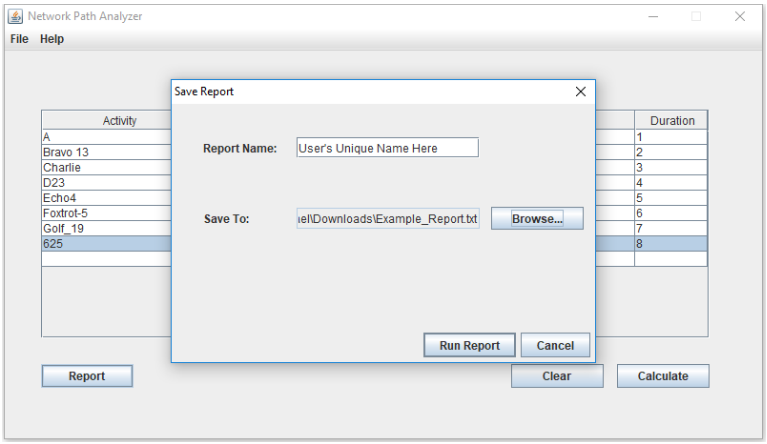
Report Name:

Save To:

Once presented with this Report pop-up, the user will be required to performed two tasks:

- Enter the name to appear on the report itself in the slot labeled "Report Name."
- Then click browse and navigate to the location on the user's computer where the report will be saved. Note that when the user arrives at the location they will have to enter a second name for the file of the report itself. This allows the user to have a different name on the report and the file of the report.

Below an example entry is shown with the name that will show on the report “User’s Unique Name Here,” and the report being saved under the name “Example_Report.”

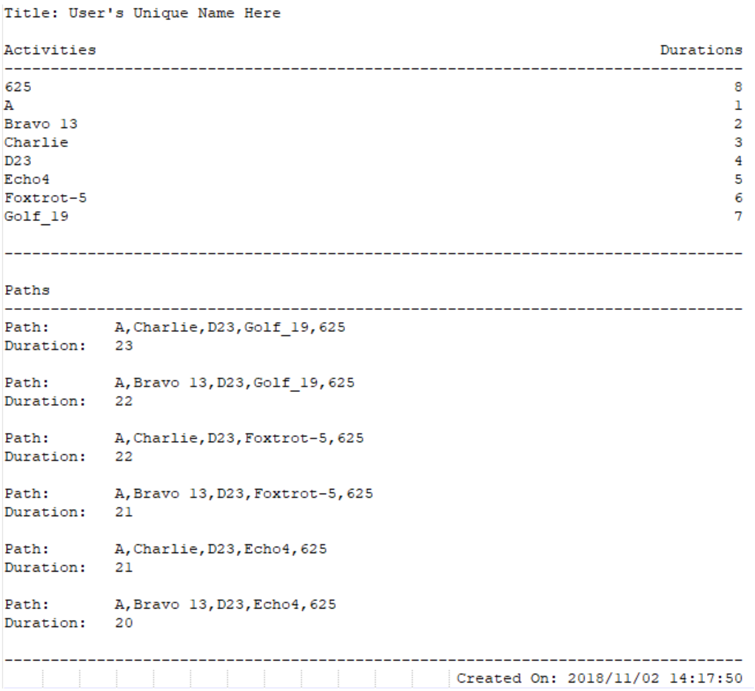


Activity	Duration
A	1
Bravo 13	2
Charlie	3
D23	4
Echo4	5
Foxtrot-5	6
Golf_19	7
625	8

Report Name: User's Unique Name Here

Save To: ıel\Downloads\Example_Report.txt

The example report saved in this instance is a text file that appears as follows when opened (Note that example name “User’s Unique Name Here” appears in upper left):

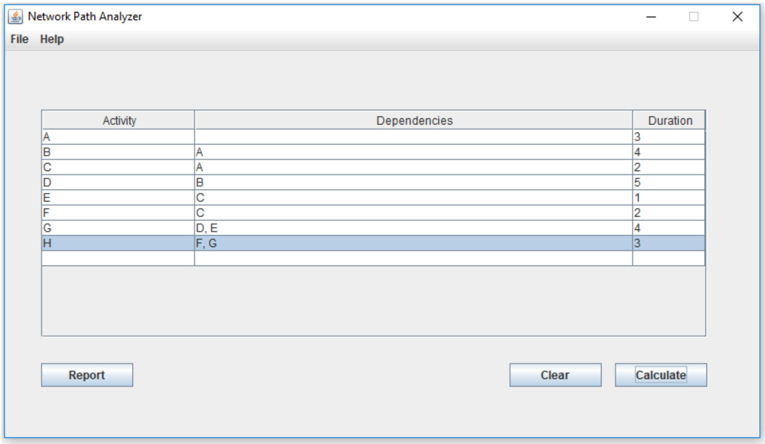


```
Title: User's Unique Name Here

Activities                                                             Durations
--------------------------------------------------------------------------------
625                                                                            8
A                                                                              1
Bravo 13                                                                       2
Charlie                                                                        3
D23                                                                            4
Echo4                                                                          5
Foxtrot-5                                                                      6
Golf_19                                                                        7

--------------------------------------------------------------------------------

Paths
--------------------------------------------------------------------------------
Path:       A,Charlie,D23,Golf_19,625
Duration:   23

Path:       A,Bravo 13,D23,Golf_19,625
Duration:   22

Path:       A,Charlie,D23,Foxtrot-5,625
Duration:   22

Path:       A,Bravo 13,D23,Foxtrot-5,625
Duration:   21

Path:       A,Charlie,D23,Echo4,625
Duration:   21

Path:       A,Bravo 13,D23,Echo4,625
Duration:   20

--------------------------------------------------------------------------------
                                                 Created On: 2018/11/02 14:17:50
```

Example Run of the Program Number 2:

Entering Path Information:

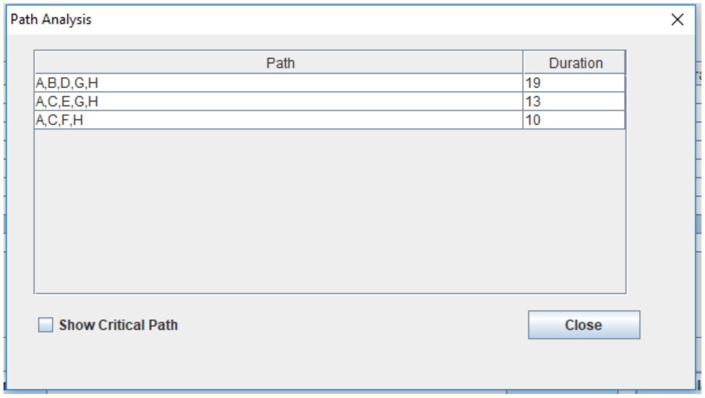
Start by entering the name of an activity (the user does not need to start with the first activity of the project and activities do not need to be entered in any particular order) into the Entry Table. Activity names can have letters, numbers, characters, and spaces (example allowable activity names: A, Charlie, D23, Foxtrot-5, Golf_19, 625). Here is a how an alternative entry of data might appear:



Activity	Dependencies	Duration
A		3
B	A	4
C	A	2
D	B	5
E	C	1
F	C	2
G	D, E	4
H	F, G	3

Calculating Paths/Output:

After entering all Activities with their corresponding Duration and Dependencies as show above the user is now ready to calculate all available paths. Click the [Calculate] button at the bottom of the screen and the Program will output the following: a list of all possible activity paths through the project, with the critical path being listed first. Paths after the critical path are listed in descending order by duration.



Path	Duration
A,B,D,G,H	19
A,C,E,G,H	13
A,C,F,H	10

□ Show Critical Path

Re-Running the Program:

To re-run the program (the user might like to do this if they want to adjust the data they have input without starting over), if the user ran the program by clicking the [Calculate] button they simply click the [Close] button of the result frame. This will close the results frame but leave the initial data entry frame open. If the user ran the program by Clicking the [Report] button, to rerun the program if the user decides not to run a report they can click the [Cancel] button on the lower right of the report pop-up and they will be returned to the Entry Table, or if they do decide to run a report once the report is generated the program will automatically take the user back to the Entry Table upon which the user has multiple options:

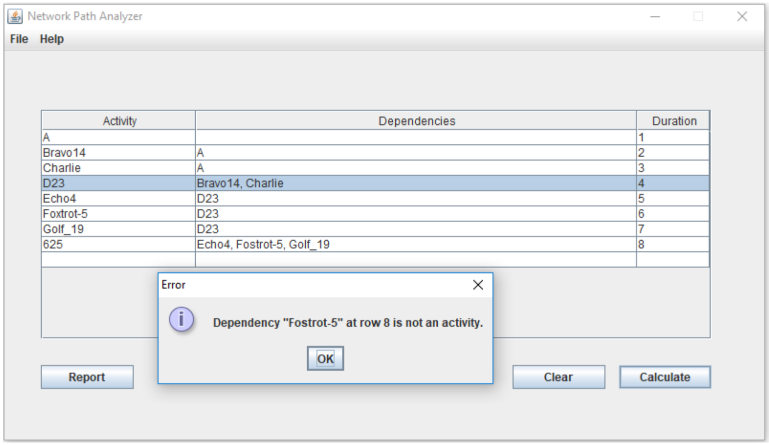
- The user can alter the data already entered by double-clicking into the cell they want to change, then deleting the old data and entering new data as needed, or
- The user can delete all the data they entered previously by clicking the [Clear] button and starting over by entering new data.

Once the user has adjusted their entered data, they can rerun the program either by clicking the [Calculate] button or the [Report] button on the data entry frame as described above. The user can continue this loop of adjusting their data and rerunning the program as many times as they like.

Examples of How Error Messages Appears:

Error if A dependency Name Does Not Match An Activity Name:

Based on Example Run of the Program 1 above, note the following example where the activity name "Foxtrot-5" is properly entered, but the reference so that activity as a dependency is mis-entered as "Fostrot-5" the error appears as follows:

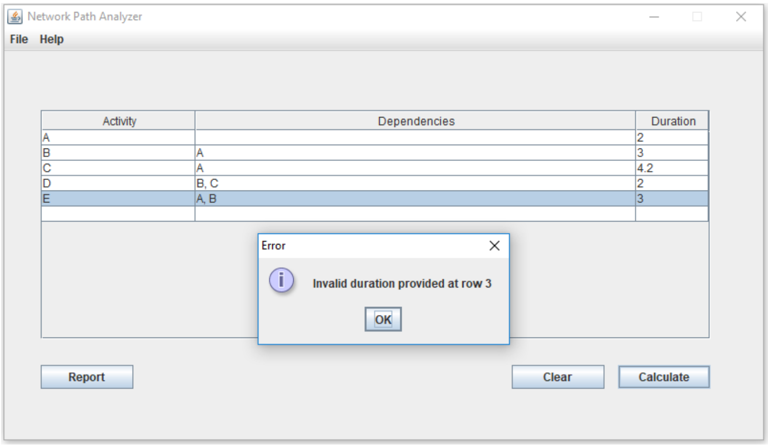


Activity	Dependencies	Duration
A		1
Bravo14	A	2
Charlie	A	3
D23	Bravo14, Charlie	4
Echo4	D23	5
Foxtrot-5	D23	6
Golf_19	D23	7
625	Echo4, Fostrot-5, Golf_19	8

Dependency "Fostrot-5" at row 8 is not an activity.

Error if a Duration is Not Entered as an Integer:

If a duration is not entered as an integer – for example as 4.2 here – the following error will display showing that the duration is invalid:

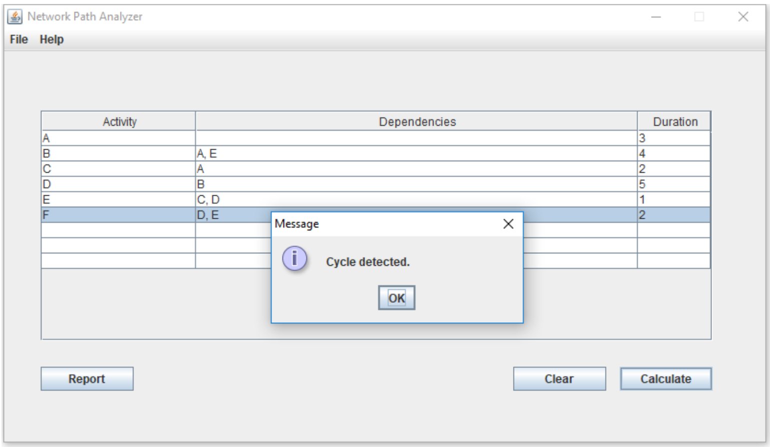


Activity	Dependencies	Duration
A		2
B	A	3
C	A	4.2
D	B, C	2
E	A, B	3

Invalid duration provided at row 3

Error if The Project as Entered Features a Cycle:

If upon analysis a project is found to contain a cycle, an error like the following will be shown:



Activity	Dependencies	Duration
A		3
B	A, E	4
C	A	2
D	B	5
E	C, D	1
F	D, E	2

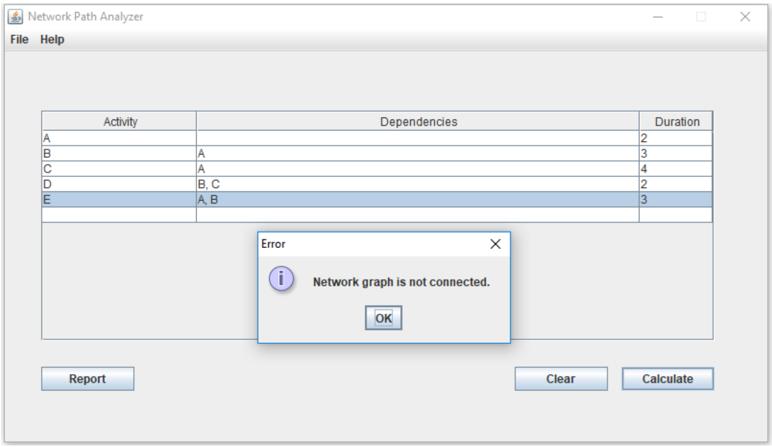
Message

Cycle detected.

OK

Error if The Project as Entered Is Not Connected:

With exception of the start activity that has no dependency or predecessor (multiple start activities are allowed), and the end activities that do not serve as a dependency or predecessor to any other activity, each of the activities of a project must both have at least one activity preceding it, and at least one activity following it. In the following data note that activity E has activities A and B as dependencies, but no other activity follows activity E. This means the project as entered is not connected and an error would be shown when the program was run (note that activity E is not a second end activity because it does not depend on activity C):



Activity	Dependencies	Duration
A		2
B	A	3
C	A	4
D	B, C	2
E	A, B	3

Error

Network graph is not connected.

OK

Conclusion:

The creators of the Network Path Analyzer are pleased to provide users with an easy, efficient way to identify critical pathways in projects of all sizes. Users enter the various activities and the program performs the mathematical organization and calculation required to find the critical path through the project with the option of generating a written report.