CHAPTER 11. CLV Launch Availability

11-0. The Problem

Perhaps the successful test of Artemis 1 in November 2022 will get us back to the moon. Well then, let's go to the moon. When shall we go? Right away. Where shall we go? Sea of Tranquility, Marius Hills, the Hadley Plains? If only it was that easy.

When launching missions to land on the moon there are two fixed parameters. The location of the launch site (for Apollo this was launch complex 39 at KSC) and the landing site you are aiming for on the moon. To arrive at this landing site with the right lighting conditions for the descent and approach a journey must be planned that meets the various constraints that affected the Apollo lunar mission planning, primarily in the form of trajectory shaping and the limitation of launch opportunities, known as launch windows.

Trajectory geometry constraints and spacecraft performance capabilities combined to limit the accessible area on the moon. These accessible landing area limitations combined with operational constraints to limit launch opportunities to certain specific periods. To fully understand the mission planning considerations and the effects of the various constraints, we would need to look closely at the trajectory characteristics. We are not going to do that here.

So, keeping it simple (or oversimplifying), let's suppose that we have a 5-day moonshot window, and we can launch any time within that window (which is really not the case but we're keeping it simple).

So, let's consider the data in ***Table 11-1*** for six major components of the upper stage of the Ares I.

Table 11-1: Major CLV Upper Stage components

Component	Repair on Launch Pad
Flight Computer	Yes
Command & Telemetry Computer	No
Data Acquisition Unit	Yes
Inertial Measurement Unit	No
Propulsion System Electronics	Yes
Reaction Control System Electronics	Yes

Simple enough. Sorry, that's too simple. There are four places where things can go wrong as the rocket is assembled, yes, some assembly required, and batteries not included). The separate pieces, the Orion Exploration Spacecraft, Upper Stage of Ares I (with Upper Stage Engine), and Lower Stage of Ares I are transported separately to the Vehicle Assembly Building (VAB). Next, the assembles space launch vehicle is moved to by a transporter along a 4-mile journey to the Launch Pad that can take between eight to 12 hours. After it leaves the VAB, the Roll-Out phase, it is paused to do some technical stuff that cannot be accomplished in the VAB (i.e., to retract the Crew Access Arm (CAA)). This is called the Stack phase. After this pause, which may be different for Artemis, the stack went to Call-to-Station before readying to be established at the Launch Pad. Due to the pauses that occur, we could say there are four points of possible failures:

1. Roll-Out
2. Stack
3. Call-to-Station
4. Launch Pad

Now, since we have some components that cannot be repaired on the Launch Pad, any failure along the way is going to cause us to return to the VAB to perform any repairs. So, regardless of any progress we have made, we now have to factor in the time for the repairs and the 4-mile trip to the Launch Pad. ***Table 11-2*** provides the delay for repair data.

Table 11-2: Repair time distributions

Event	Purpose	Distribution	Parameter 1	Parameter 2
Repair Flight Computer	TTU	Normal	1	360
Repair Command & Telemetry Computer	TTU	Normal	42	13
Repair Data Acquisition Unit	TTU	Normal	2	1
Repair Inertial Measurement Unit	TTU	Normal	42	13
Repair Propulsion System Electronics	TTU	Normal	2	1
Repair Reaction Control System Electronics	TTU	Normal	2	1

To keep this scenario from getting out of control, let's "assume away" the time it takes to travel between VAB and Launch Pad and just consider the repair time. In ***Table 11-2***, we've added additional delay time for repair of components not repairable on the Launch Pod.

11-1. Reliability Diagrams

Now, we have enough information to map out the problem with a **reliability diagram**. This kind of diagram is drawn to depict events that are redundant, i.e., each downing event for one component does not affect the other or there is an existing backup component. These are drawn as a vertical stack and the CLV can continue to the Launch Pad and be repaired there. On the other hand, a component downing event can affect the other components, i.e., a failure of the Inertial Measurement Unit causes us to return the entire stack to the VAB. These components are placed on the diagram as a horizontal series. When one fails, we place a return flow arrow to the VAB. ***Figure 11-1*** represents the reliability diagram for our problem. Notice that in our case the values R_1 to R_6 are not reliabilities. Instead, are using repair times assigned by ***Table 11-2***.

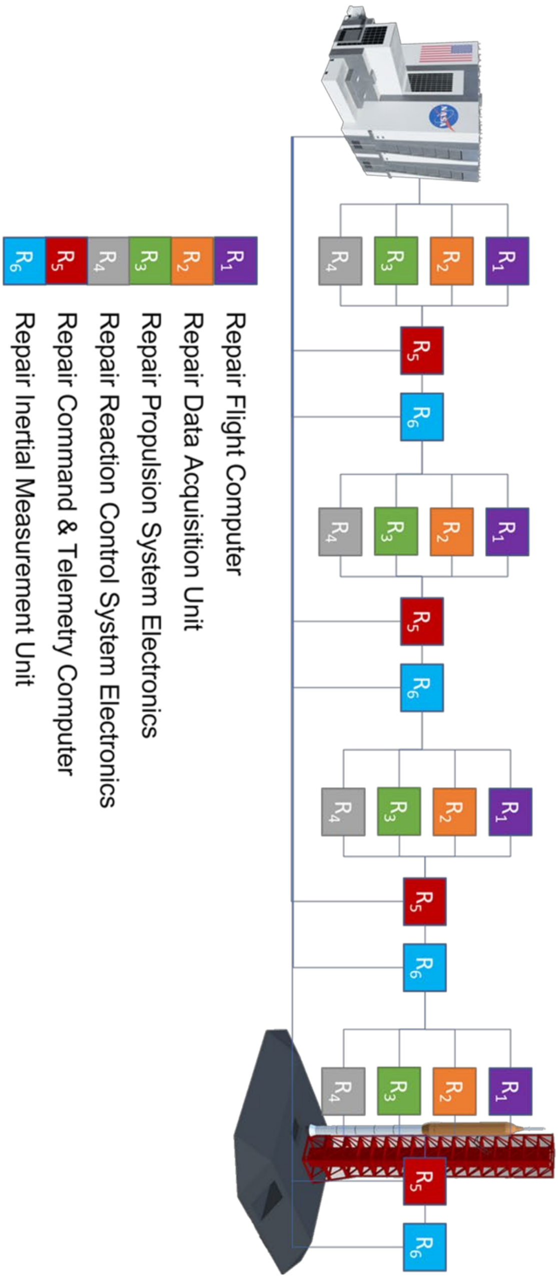


Figure 11-1: Roll-Out to Land Pad Ares I reliability diagram

Figure 11-2 shows additional details including the distributions of repair times, R_i. The "official" term for repair time is **Mean Time to Repair** (MTTR), or **Time to Up** (TTU).

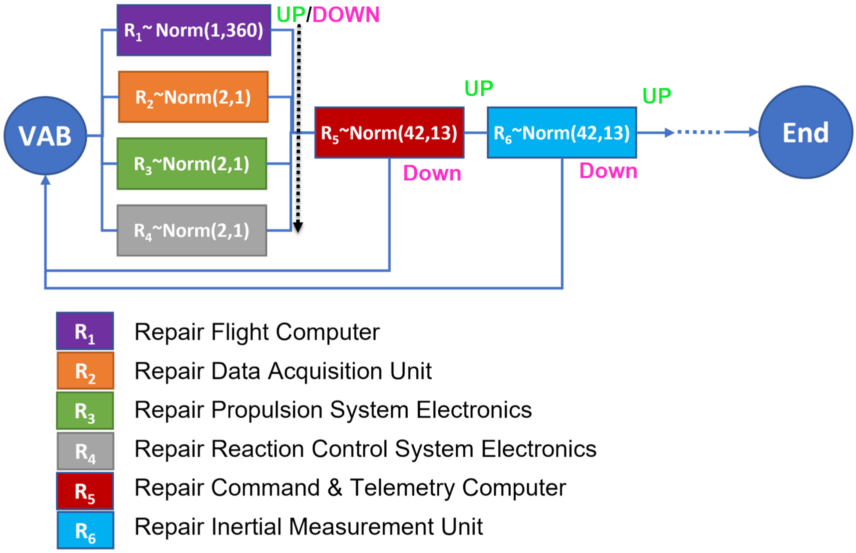


Figure 11-2:Reliability diagram at one of the four "stations"

There are two more pieces of information we need to model our problem. Table 11-3 provides the probability distributions for Time Before Down (TBD), Time To Down (TTD) and Time To Up (TTU). Notice that the Annual Maintenance and Bi-Annual Maintenance uses Time Between Downs. We are only using these in the model to reduce period of wear and tear to 6 months. It's like the preventive maintenance you do on your car.

Table 11-3: Distribution Table for the CLV Launch Availability

Name	Pur-pose	Distri-bution	Param-eters 1	Param-eter 2	Time Units
Annual Maintenance	TBD	Constant	525600	0	Hours
Bi-Annual Maintenance	TBD	Constant	262800	0	Hours
Failure Flight Computer	TTD	Weibull	6000	51000	Hours

Failure Command & Telemetry Computer	TTD	Weibull	4250	5150	Hours
Failure Data Acquisition Unit	TTD	Weibull	2000	6000	Hours
Failure Inertial Measurement Unit	TTD	Weibull	4000	5900	Hours
Failure Propulsion System Electronics	TTD	Weibull	4000	5000	Hours
Failure Reaction Control System Electronics	TTD	Weibull	4200	5100	Hours
Annual Maintenance	TTU	Constant	24	0	Hours
Bi-Annual Maintenance	TTU	Constant	24	0	Hours
Repair Flight Computer	TTU	Normal	1	360	Hours
Repair Command & Telemetry Computer	TTU	Normal	42	13	Hours
Repair Data Acquisition Unit	TTU	Normal	2	1	Hours
Repair Inertial Measurement Unit	TTU	Normal	42	13	Hours
Repair Propulsion System Electronics	TTU	Normal	2	1	Hours
Repair Reaction Control System Electronics	TTU	Normal	2	1	Hours

Table 11-4 provides the classes for the events through which the distributions are assigned. Recall that our software availability model had a **Distribution Builder** and an **Event Builder** that used these two tables.

Table 11-4: Class Table for the CLV Launch Availability

Event Name	Event Type	TBD/TTD: Dist ID	TBD/ TTD: Progress Type	TBD/ TTD Prob > 0	TTU: Dist ID	TTU: Progress Type
Annual Maintenance	Distribution	TBD-Annual Maintenance	Time	1	TTU-Annual Maintenance	Time
Bi-Annual Maintenance	Distribution	TBD-Bi-Annual Maintenance	Time	1	TTU-Bi-Annual Maintenance	Time
Flight Computer	Distribution	TTD-Flight Computer	Time	1	TTU-Repair SP Propagator	Time

Command & Telemetry Computer	Distribution	TTD-Command & Telemetry Computer	Time	1	TTU-Repair Obs VAR	Time
Data Acquisition Unit	Distribution	TTD-Data Acquisition Unit	Time	1	TTU-Repair ObsProc Metrics	Time
Inertial Measurement Unit	Distribution	TTD-Inertial Measurement Unit	Time	1	TTU-Repair OMR	Time
Propulsion System Electronics	Distribution	TTD-Propulsion System Electronics	Time	1	TTU-Repair PCT	Time
Reaction Control System Electronics	Distribution	TTD-Reaction Control System Electronics	Time	1	TTU-Repair Retag Priority	Time

11-2. Building the Model

To build the CLV Launch Availability Model, we start with a new, blank worksheet and we name it Launch Availability 1.mox. Next, we use our reliability diagrams to place reliability nodes on the worksheet. We provide a possible finished model picture in ***Figure 11-3*** but you'll have to get there on your own (mostly).

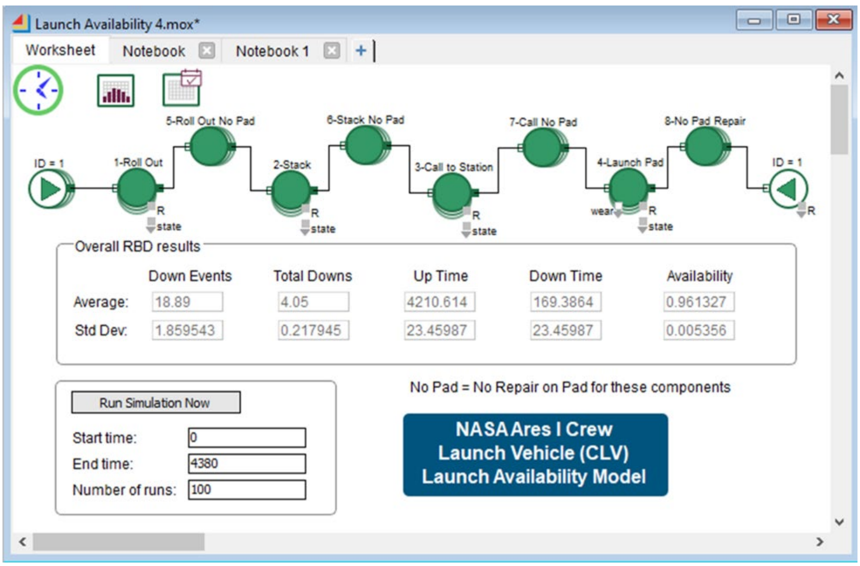


Figure 11-3: A version of the completed CLV Launch Availability model

We do, however, provide these steps:

Step 1: From the Reliability Library place a Start Node on the new Worksheet. When prompted with **Ask for component names?**, as seen in , select **Yes**.

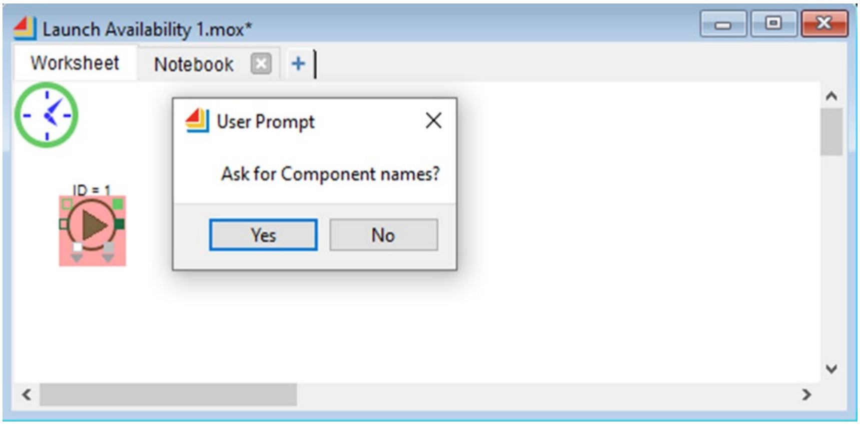


Figure 11-4: Prompt for adding component names

Step 2: Add component to the worksheet as shown in ***Figure 11-5***.

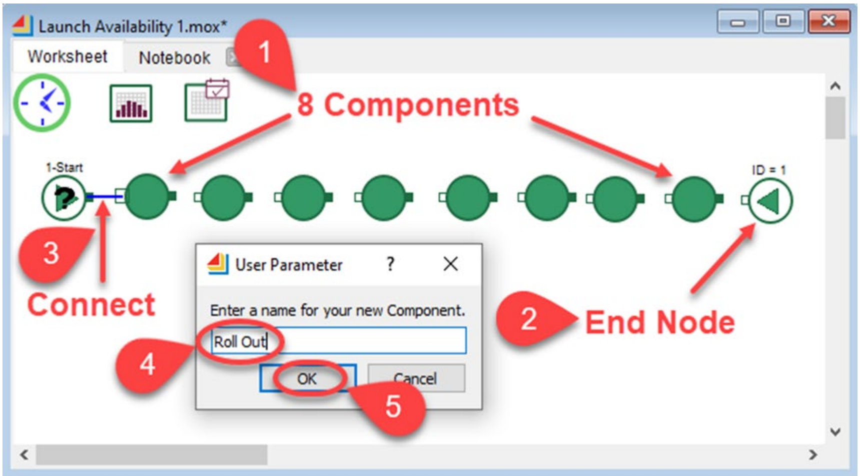


Figure 11-5: Component node placement on the model worksheet

Step 3: After all the nodes have been added to the worksheet and connected, you should name them as we did with the Software Availability model, according to ***Figure 11-6***.

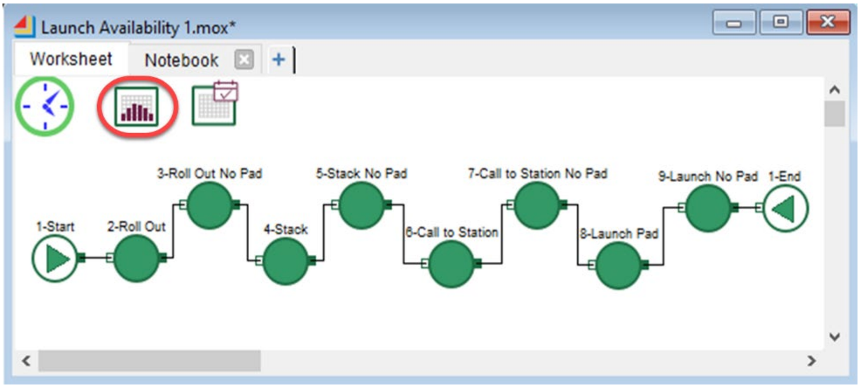


Figure 11-6: Node naming

Step 4: Use the **Distributions Builder** to define the distributions for TBD/TTD and TTU. On the **Distribution** tab, as shown in ***Figure 11-7***.

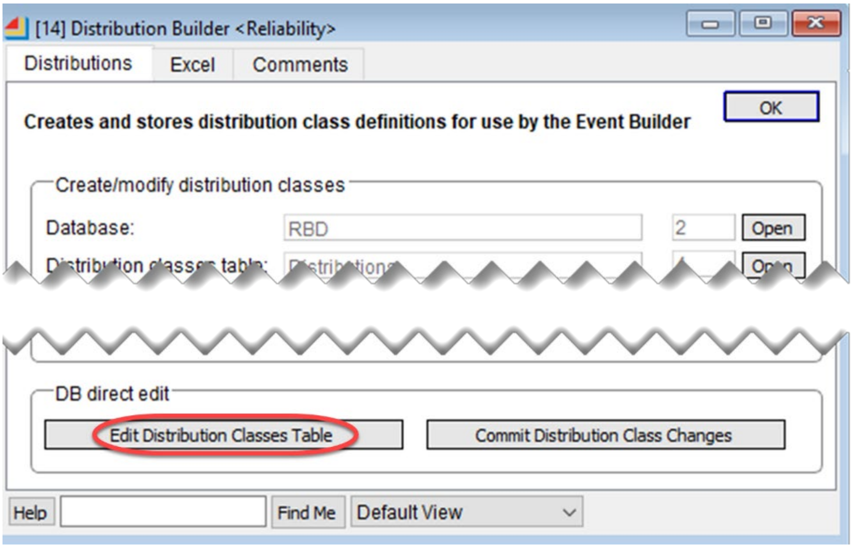


Figure 11-7: Selecting the distributions classes table for editing

Step 5: When the Distribution table is opened, it should be blank except for the headings. So, add 16 rows to the **Distribution** table, as shown in ***Figure 11-8***.

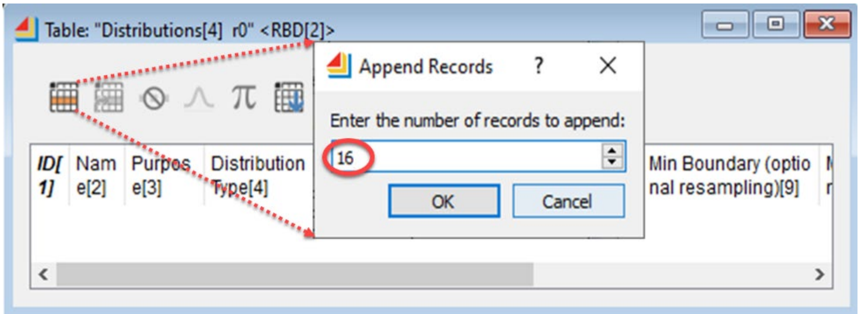
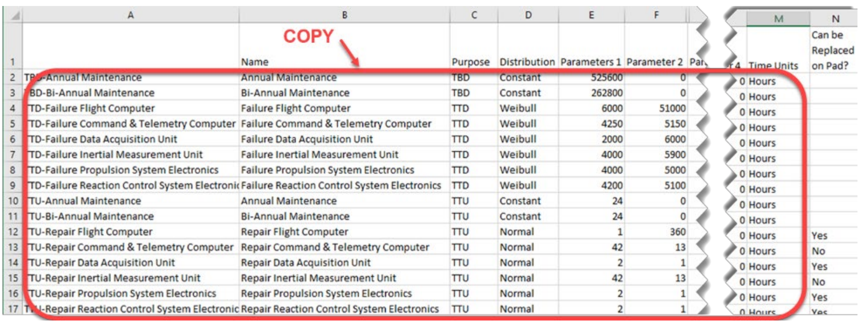


Figure 11-8: Adding rows to the Distribution table

Step 6. Copy the Distributions from the Excel file, CLV Availability.xlsx as shown in ***Figure 11-9***, and copy them to the Distribution table as shown in ***Figure 11-10***.



	A	B	C	D	E	F	L	M	N
1		Name	Purpose	Distribution	Parameters 1	Parameter 2		Time Units	Can be Replaced on Pad?
2	TB-Annual Maintenance	Annual Maintenance	TBD	Constant	525600	0	0	Hours	
3	BD-Bi-Annual Maintenance	Bi-Annual Maintenance	TBD	Constant	262800	0	0	Hours	
4	TD-Failure Flight Computer	Failure Flight Computer	TTD	Weibull	6000	51000	0	Hours	
5	TD-Failure Command & Telemetry Computer	Failure Command & Telemetry Computer	TTD	Weibull	4250	5150	0	Hours	
6	TD-Failure Data Acquisition Unit	Failure Data Acquisition Unit	TTD	Weibull	2000	6000	0	Hours	
7	TD-Failure Inertial Measurement Unit	Failure Inertial Measurement Unit	TTD	Weibull	4000	5900	0	Hours	
8	TD-Failure Propulsion System Electronics	Failure Propulsion System Electronics	TTD	Weibull	4000	5000	0	Hours	
9	TD-Failure Reaction Control System Electroni	Failure Reaction Control System Electronics	TTD	Weibull	4200	5100	0	Hours	
10	TU-Annual Maintenance	Annual Maintenance	TTU	Constant	24	0	0	Hours	
11	TU-Bi-Annual Maintenance	Bi-Annual Maintenance	TTU	Constant	24	0	0	Hours	
12	TU-Repair Flight Computer	Repair Flight Computer	TTU	Normal	1	360	0	Hours	Yes
13	TU-Repair Command & Telemetry Computer	Repair Command & Telemetry Computer	TTU	Normal	42	13	0	Hours	No
14	TU-Repair Data Acquisition Unit	Repair Data Acquisition Unit	TTU	Normal	2	1	0	Hours	Yes
15	TU-Repair Inertial Measurement Unit	Repair Inertial Measurement Unit	TTU	Normal	42	13	0	Hours	No
16	TU-Repair Propulsion System Electronics	Repair Propulsion System Electronics	TTU	Normal	2	1	0	Hours	Yes
17	T-Repair Reaction Control System Electronic	Repair Reaction Control System Electronics	TTU	Normal	2	1	0	Hours	Yes

Figure 11-9: Excel workbook with distributions

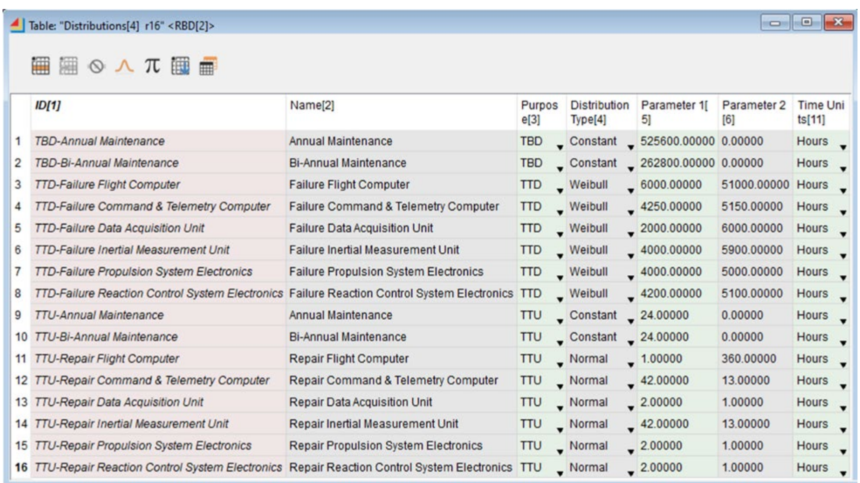


Table: "Distributions[4] r16" <RBD[2]>

	ID[1]	Name[2]	Purpose[3]	Distribution Type[4]	Parameter 1[5]	Parameter 2[6]	Time Units[11]
1	TBD-Annual Maintenance	Annual Maintenance	TBD	Constant	525600.00000	0.00000	Hours
2	TBD-Bi-Annual Maintenance	Bi-Annual Maintenance	TBD	Constant	262800.00000	0.00000	Hours
3	TTD-Failure Flight Computer	Failure Flight Computer	TTD	Weibull	6000.00000	51000.00000	Hours
4	TTD-Failure Command & Telemetry Computer	Failure Command & Telemetry Computer	TTD	Weibull	4250.00000	5150.00000	Hours
5	TTD-Failure Data Acquisition Unit	Failure Data Acquisition Unit	TTD	Weibull	2000.00000	6000.00000	Hours
6	TTD-Failure Inertial Measurement Unit	Failure Inertial Measurement Unit	TTD	Weibull	4000.00000	5900.00000	Hours
7	TTD-Failure Propulsion System Electronics	Failure Propulsion System Electronics	TTD	Weibull	4000.00000	5000.00000	Hours
8	TTD-Failure Reaction Control System Electronics	Failure Reaction Control System Electronics	TTD	Weibull	4200.00000	5100.00000	Hours
9	TTU-Annual Maintenance	Annual Maintenance	TTU	Constant	24.00000	0.00000	Hours
10	TTU-Bi-Annual Maintenance	Bi-Annual Maintenance	TTU	Constant	24.00000	0.00000	Hours
11	TTU-Repair Flight Computer	Repair Flight Computer	TTU	Normal	1.00000	360.00000	Hours
12	TTU-Repair Command & Telemetry Computer	Repair Command & Telemetry Computer	TTU	Normal	42.00000	13.00000	Hours
13	TTU-Repair Data Acquisition Unit	Repair Data Acquisition Unit	TTU	Normal	2.00000	1.00000	Hours
14	TTU-Repair Inertial Measurement Unit	Repair Inertial Measurement Unit	TTU	Normal	42.00000	13.00000	Hours
15	TTU-Repair Propulsion System Electronics	Repair Propulsion System Electronics	TTU	Normal	2.00000	1.00000	Hours
16	TTU-Repair Reaction Control System Electronics	Repair Reaction Control System Electronics	TTU	Normal	2.00000	1.00000	Hours

Figure 11-10: Distribution table in ExtendSim.

Step 7. Close the tables and make sure you **Commit** the additions/ changes, as shown in ***Figure 11-11***.

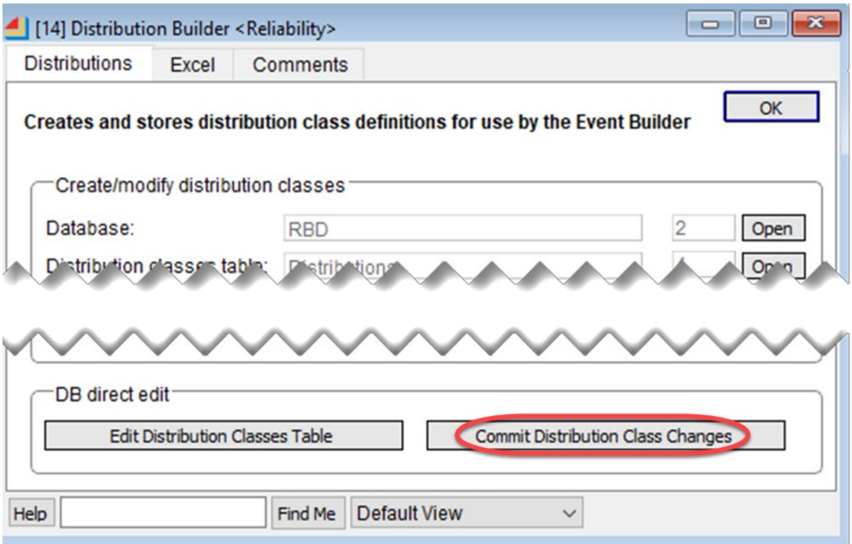


Figure 11-11: Committing distribution class changes

Step 8: Open the **Event Builder** dialog (see ***Figure 11-12***) to define the events and for assigning their respective distributions.

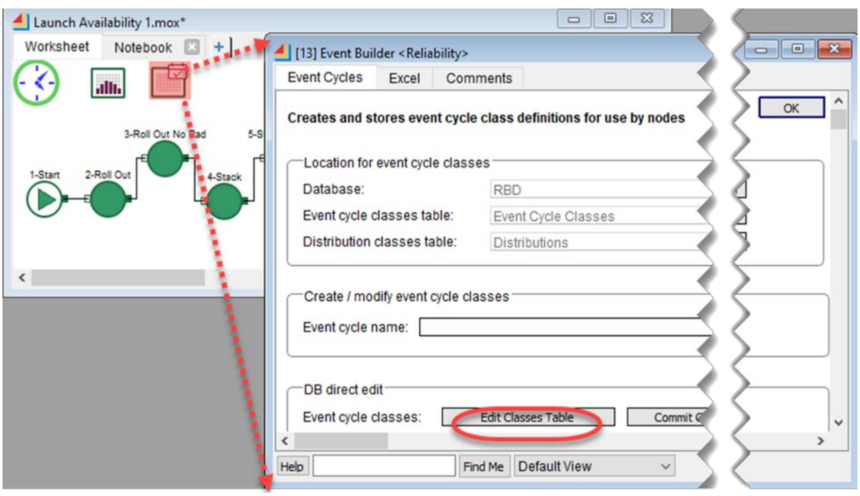


Figure 11-12: Event Builder dialog window

Step 9. Copy the event classes from the Excel spreadsheet (**CLV Availability.xlsx**) to the **Event Cycle Classes** table. And select the distributions for TBD/TTD and TTU.

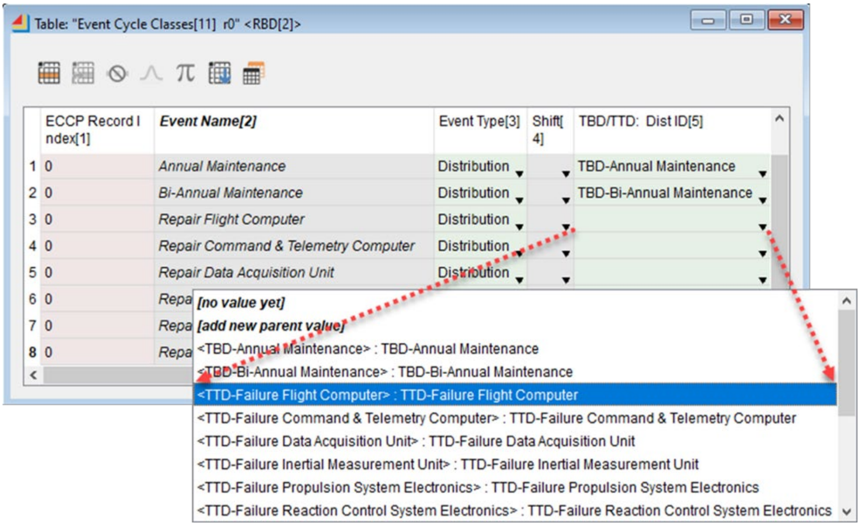


Figure 11-13: Selecting event classes dstributions

Step 10. Close the table and commit the changes as shown in *Figure 11-14*.

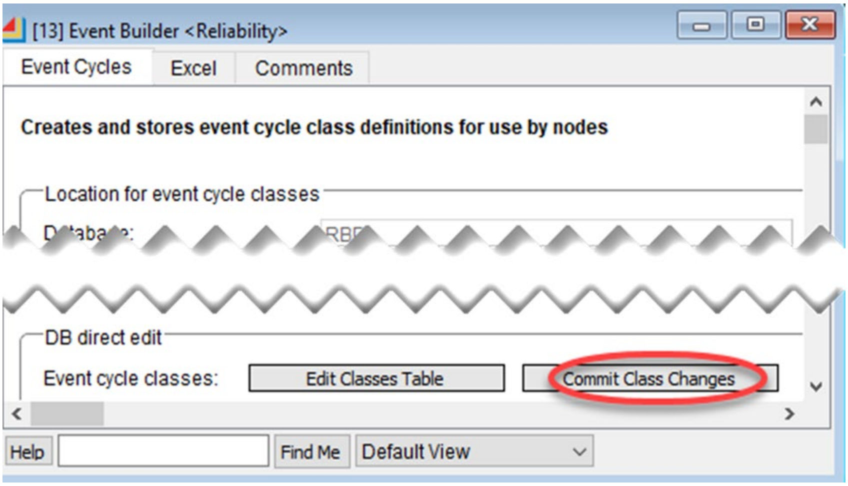


Figure 11-14: Committing class changes

Step 11: Using the Start Node, assign the event classes for each node in the model. For the Start node, we provide an illustration with ***Figure 11-15***. Why did we enter **Ignore** for the interrupts? Is the **SN-DE Interrupt** missing? See **7** in the figure. Remember that we need values (**Preserve**, **Reset**, or **Ignore**) for every SN-DE Interrupt.

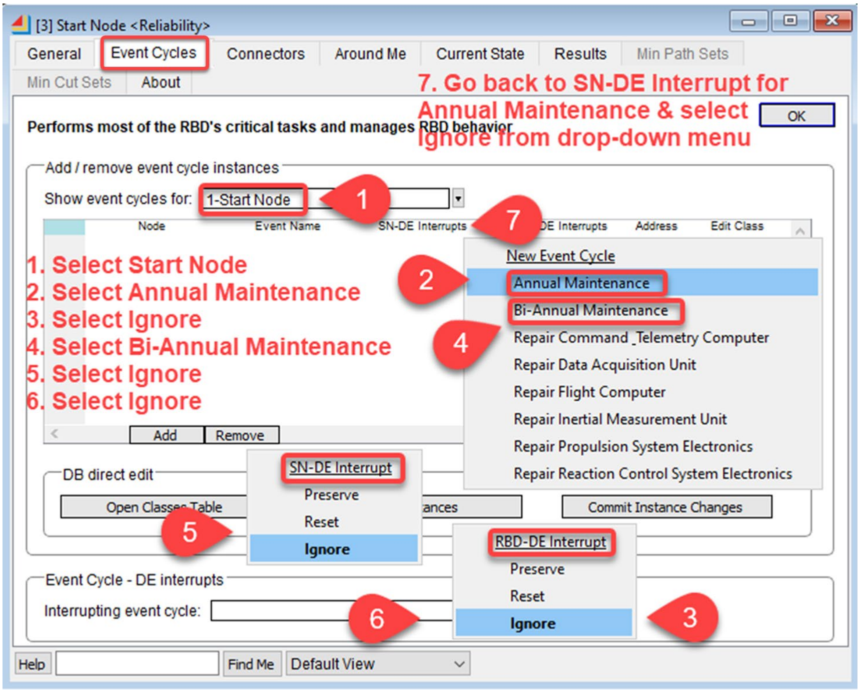


Figure 11-15: Assigning classes to the Start Node

Take notice of the **Stack node** settings in ***Figure 11-16***. All of the **Interrupt** settings are **Preserve** and the **Event cycles placed in:** is set to **parallel**.

Step 12: Repeat **Step 11** for all nodes.

Step 13: Ensure you view and then **Commit Instant Changes**. This is a step most people miss, and the model will not run unless you perform this step. See ***Figure 11-17***. No entries are required.

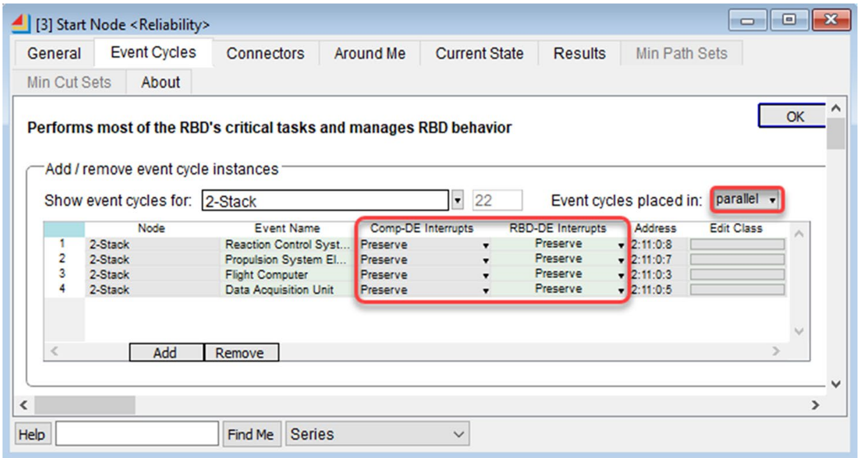


Figure 11-16: Settings for the Stack Node

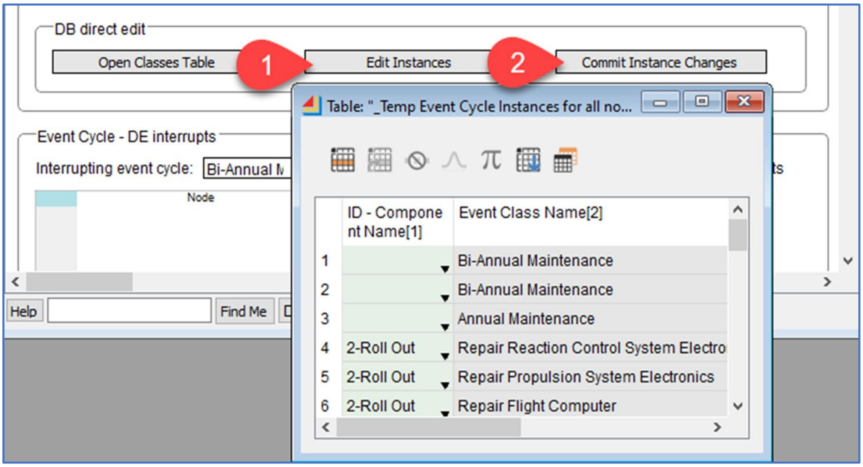


Figure 11-17: Committing Instances

Step 14: Set up the simulation. Use 4380 hours (six months) and 100 runs.

Step 15: Run the simulation to check it is functioning as expected (refer back to ***Figure 11-3***).

Step 16: If Step 8 fails, trouble-shoot the model. A solution is available on my **Github** site as noted in the preface.

Step 17: Clone and add the results table to the worksheet (**Results** are in the **Start Node**).

Step 18: Explain your solution.

Step 19. Add/clone the simulation setup to the worksheet as shown in ***Figure 11-6.***

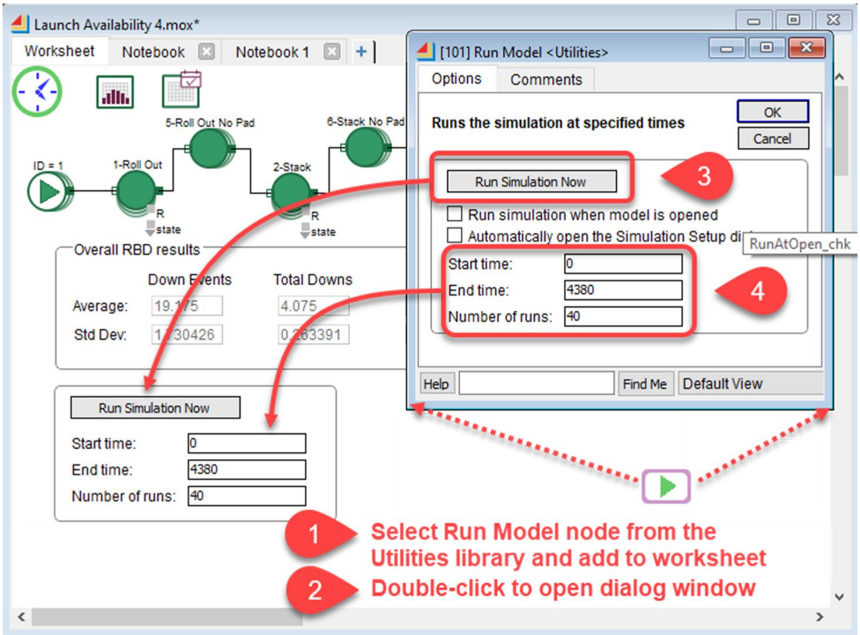


Figure 11-18: Cloning the simulation setup to the worksheet

11-3. Afterword

The **Reliability Library** is very efficient. If we attempted this model with the **Item Library**, it would have used more nodes and may have been more complicated.