

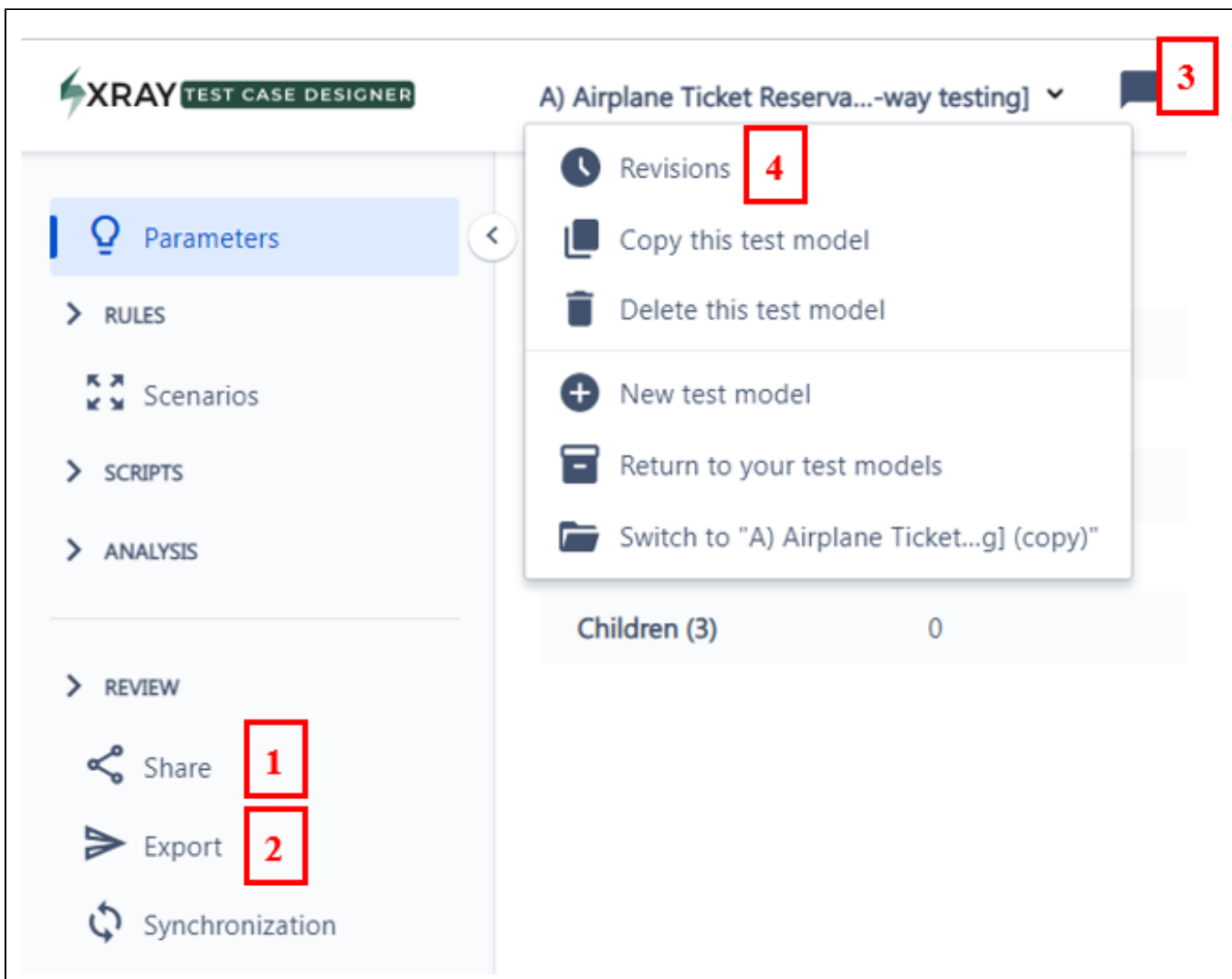
Sharing & Maintenance Tips

When teams start exploring Test Case Designer, they are often unsure what will happen to the models after the initial release – will they have long-term value, where is traceability with execution, how to conduct maintenance, etc. ? In this article, we have consolidated the “cheat sheet” from the information available in our feature-focused documents and provided additional insights gained from client projects.

- [Part 1. Collaboration and review](#)
- [Part 2. Release preparation and ongoing maintenance](#)
 - [Freezing](#)
 - [Ongoing maintenance](#)
 - [Simple Changes](#)
 - [Complex Changes](#)

Part 1. Collaboration and review

As you design TCD models, it is always a good idea to work together with other SMEs, bounce ideas off business analysts and developers, and eventually review the draft models with the key stakeholders. To assist in that process, TCD provides several features:



1. TCD models can be shared at the project level by separately including email addresses or sharing the secret link with the group. You can choose one of 3 permission levels for each colleague. If you do not want some models to be seen, you can move them from the project state to the private one by editing in the “Your Test Models” dialog.



Sharing also ensures that models do not get lost when people transition to different roles or move to other job opportunities.

2. While sharing enables the most interactive access to TCD capabilities, sometimes export is more practical. Some of the most popular collaboration formats include mind maps, Excel, and feature files.
3. Notes can be used to keep track of your design ideas, to post questions to your collaborators, or to mark outstanding actions to complete the model.
4. Revisions (accessed from the dropdown near the model name) provide version control capabilities where you can see your colleagues' actions and, if necessary, return the model to the previous state.

Part 2. Release preparation and ongoing maintenance

Once you have created the optimal model and confirmed it with the stakeholders, it is time to move the test suite to execution. Many downstream actions are typically performed in test management tools (e.g., Xray) or automation frameworks (e.g., Ranorex Studio). We will focus on the steps that have a direct impact on the Test Case Designer models.

Freezing

When your scenario suite is finalized, it can be a good idea to ensure test data consistency with the help of "freezing".

14 scenarios and 154 2-way interactions

Flying to	Class	Seat Preference	Adults	Children	Amount of luggage
the United States	Coach	None	1	0	0
the Philippines	Business	Selected	More than 1	0	1
India	First	Selected	1	0	2 or more
India	Business	None	More than 1	1	0
the United States	Coach	Selected	More than 1	1	1
the Philippines	First	None	1	More than 1	0
India	Coach	None	1	More than 1	1
the United States	Business	None	More than 1	More than 1	2 or more
the Philippines	Coach	None	1	1	2 or more
the United States	First	None	More than 1	1	1
<i>the United States</i>	Business	Selected	1	More than 1	0
<i>the Philippines</i>	<i>Coach</i>	<i>Selected</i>	<i>More than 1</i>	<i>1</i>	0
<i>the Philippines</i>	<i>Coach</i>	<i>Selected</i>	<i>More than 1</i>	<i>1</i>	1
<i>the United States</i>	<i>First</i>	<i>None</i>	<i>1</i>	<i>0</i>	2 or more

Freeze these Scenarios

To clarify, freezing makes the italicized values (red rectangle highlighted above) constant by saving all scenarios as Forced Interactions. Italic type means that TCD has already covered all n-way interactions for that parameter and now randomly selects a value from the available list. Therefore, without freezing, there is a chance the value will change on suite re-generation. This could pose a challenge with test data preparation if, for example, distinct accounts must be set up to represent each scenario.

Ongoing maintenance

While we recommend unfreezing the test suite before incorporating new information into the model, in some cases, the updates can be made on top of the preserved scenarios. You have to remember that frozen test cases will often prevent TCD from finding the most efficient way of including changes, which leads to the avoidable growth of the test suite.

We will focus on the situation where the test suite can be re-generated completely. Regarding sources of information, new requirements are the ones mentioned most often, but you should not overlook execution and defect reports. While TCD does not execute your tests, establishing the feedback loop and ensuring your TCD model includes the interactions that have caused issues in the system is critical for the long-term value of the tool.

Simple Changes

This category includes renaming values, adding/removing values that do not affect constraints, and changing the script.

- To avoid unnecessary complications when you just want to rename an individual Value, we recommend you do that by clicking on the actual Value, not the parameter name.



- Adding or removing values can be done via Edit, Bulk Edit, or Mind Map modes – depending on the number and format of changes.
- Within Auto-scripts, you can edit the content of each step and its position in the overall script by hovering over and clicking the pencil/arrow icons:



Complex Changes

We will look at 2 examples: significant updates to the constraint logic and merging ideas from multiple models.

Example 1: When the input set is constant, and the constraint logic changes, you have to either delete relevant constraints 1-by-1 or, if the updates are drastic, bulk delete them and start from scratch.

- You can leverage the browser search to identify constraints for the necessary values:

A) Airplane Ticket Reservation 1

business 1/2

Delete All

never	Flying From = India	Flying From (3)	✗ 1 India	✗ 1 the Philippines	✗ 1 the United States
never	Flying to = India	Flying to (3)	✗ 1 the United States	✗ 1 the Philippines	✗ 1 India
never	Flying From = the Philippines	Class (3)	Coach	✗ 1 Business	First
never	Flying to = the Philippines	Seat Preference (2)	None	Selected	
never	Flying From = the United States	Adults (2)	1	More than 1	
never	Flying to = the United States	Children (3)	0	1	✗ 1 More than 1
never	Class = Business	Amount of luggage (3)	0	1	2 or more
never	Children = More than 1				

When the parameter set changes and affects the constraint logic, the warnings will appear as you adjust the content on the first TCD tab. Please read through the impacted constraints and ensure the change you are making is the correct & efficient one. If you see a banner about “No possible values,” refer to another article in this category.

Parameter Update Warning

✕

Updating **Flying to** will delete **1 constraint**.

If you save now, updating **Flying to** will change **1 requirement** by deleting affected parameter and parameter value conditions.

Are you sure you want to update **Flying to**?

Cancel

Save Anyway

Example 2: When you need to incorporate ideas from multiple TCD models into one, you need to select the core model which has the highest amount of complexity (constraints, auto-scripts, etc.). You will copy that model as a whole through the “Your Test Models” dialog and then add content from the other models.

- You can add parameters via copy-paste in Bulk Edit or via parameter lookup in the “New Parameter” dialog:

New Parameter

Parameter Name

Temperature (4)

Term of Loan (3)

TicketType (3)

Transaction Exchange (Country) (14)


Trip Information (3)

Type of Animal (3)

Type of Flight(2)

Cancel

Create

 Parameter lookup works at the project level.

- You can add constraints via copy-paste in Advanced Mode
- Depending on the elements of each model, it may be more efficient to merge 3 exports in Excel and create a new model from Excel import:

Importing Plans from Files



Import from Excel

Provide an Excel spreadsheet with a sheet named **Inputs**. Each column in the sheet is a parameter and each row has parameter values:

Animals	Letters	Parameter 3
Monkey	A	Value 1
Ferret	B	Value 2
	C	

The spreadsheet must be in the **Excel Workbook** file format and end in a **.xlsx** extension. Save a spreadsheet in Excel Workbook format by selecting **File→Save As** and picking the **.xlsx** file type.

To ensure you have the correct file format for importing, export an existing plan, or start with [this Excel example](#) file. If you follow the format from an export or this sample, you can also import constraints, value expansions, notes, and auto-scripts from their respective sheets.

Import from a Mind Map or Outline

Many [mind mapping](#) and [outlining](#) tools support exporting to **OPML**, a simple hierarchical format that can be used to define test plan inputs only. Here is a [partial list of tools](#). To import from OPML, ensure your file name has a **.opml** extension.



If you are dealing with integration testing, we often recommend decreasing the level of detail – in other words, do not merge all parameters & values from multiple models, but instead reevaluate which are the crucial ones from the integration perspective and abstract the rest.