Creating a Child Case :**-**

* Child case is something which represent work that must be completed to resolve the parent. A parent case that creates multiple child cases allows for work to be processed in parallel.
* Complex business transactions may require the creation of multiple cases that’s why we’ll create child cases.
* Child cases can be created to model work that are performed by different parties or for work that follows different reporting options

**Child cases are beneficial in situations where you want to**:

* Model work that different parties perform
* Model work that follows different reporting options
* Model work that follows different time frames
* Model work that is common to multiple case types
* Child cases are also beneficial in situations for modeling work separately from the parent case when several of the items in the list below are true:
* A different data model is needed
* A different life cycle is needed
* A separate case ID and status for reporting are necessary
* When separately assigning the case is needed
* It must complete before the parent completes

**Case relationships: Top-level case and child case**

* **Top-level case** a case that does not have a parent case but can become a parent of other cases.
* **Parent case** creates multiple child cases that allow for work to be processed in parallel.
* **Child case** represents work that must be completed to resolve the parent.

**How to Propagate data from child to parent ?**

1. Refer to data page.

2. Copy data from a data page

3. Data transform

4. Directly from Data propagation ( you can find it case lifecyle --> Settlings --> Data propagation)

**1.What is Case locking strategy?**

~To avoid data corruption or loss due to overwrites we use this strategy.

………(When two or more actions attempt to update a case, the last-performed action may overwrite data written by a previous action. Overwrites can lead to the corruption or loss of data, delaying case processing, and potentially leading to the incorrect resolution of the case. If an application supports concurrent users, an appropriate case locking strategy is essential to ensuring data integrity.)

**2.What are the Types of Locking Strategies ?**

~**pessimistic locking** and **optimistic locking**.

**3.What is pessimistic locking**?

~ an application applies an exclusive lock when opening an item, such as a case. While the item remains locked, other users cannot edit the item.

-after completion of editing the lock is removed and it is free to edit.

- Pessimistic locking ensures data integrity by limiting case access.

**4.What is optimistic locking**?

~ With an **optimistic locking** strategy, an application does not apply an exclusive lock when opening an item. Instead, any user — or the system itself — can open and edit the item at any time. When the item changes, the application checks whether the item has changed before committing any changes.

- User 1 opens the item and begins editing.

- User 2 opens the item. Because the application does not lock the item for User1 , User 2 can edit the item.

- User 2 saves the item and commits their changes.

- User 1 attempts to save the item. The application informs User 1 that the item is stale because User 2 already committed their changes.

5.What is instantiation ?

~  When a new case is created, it is said to be **instantiated.**

Parent cases are usually instantiated in the Create menu on the Case Manager portal. Subcases are instantiated in the following ways:

* Automatically — By the system when the parent case is created.
* Manually — By the user when the case type is selected from the Other Actions ... menu in the parent work form.

# **Automatic Instantiation:**

To enable automatic instantiation, select the **Automatically by system when** check box and choose one of the following options:

* **the parent case starts** — The subcase begins when the parent case type is instantiated.  
  Optionally, enter a when rule in the **only if**[rule]**evaluates to true** field.
* **All/Any of the following dependencies are fulfilled** — The subcase begins when the settings defined in the **Dependencies** item for this case type are met. Choose rules and conditions (**has started**, **has work status**, **has completed**) in the fields below.

# **Manual Instantiation:**

To enable manual instantiation, select **Manually by user**. Optionally,select**only when [rule] evaluates to true** and enter a rule in the field. When it evaluates to true, the user can create a subcase.

**6. What is Data Propagation?**

~ Data propagation is the process of mapping run-time values of the properties in a parent case to properties in a child case when the child case is created. By sharing information among cases, you can make data-driven decisions.

**CONFIGURATIONS:**

**creating multiple child cases using list**

**https://youtu.be/g20B163dfao?si=oWGvzP\_3pJ8afqdB**

* Click dropdown for detailed steps

# Creating a child case type

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the process in which you want to add a child case type, click **Add step to the process** **More** **Automations** **Create case** , and then click **Select**.
3. In the **Create case** field, enter the name of the child case type.
4. On the **Step** tab, in the **Create the following case** list, select a type of a child case type that you want to create:
   * To create a new child case type, select **Create new case type**, , enter a unique name for the case type, and then click **Create**.
   * To reuse an existing case type, select the case type.
5. **Optional:**To create more child case types, select **Create multiple cases using a list**, and then go to step [4](https://academy.pega.com/topic/creating-child-case-type/v1#Creating_a_child_case____type/step_qrt_vmb_h3b).
6. **Optional:**To reuse information from another case, select the **Transfer information to new case** check box, and then, in the **Transfer information** window, select the information that you want to transfer:
   * In the **'From' field** column, select the source field that stores the information that you want to transfer.
   * In the **'To' field** column, in the text box that appears, enter the field in your child case where you want to use the information.
   * **Optional:**To add mapped fields to the case type view, in the **View** section, select the **Add mapped fields to *[case type view name]* view** check box.
   * Click **OK**.
7. Click **Save**.

# **Configuring parent-child instantiation for a case type**

You can configure a child case type so that it is automatically instantiated when its parent case type is instantiated. By allowing your application to manage run-time dependencies, you can reduce the workload for case workers.

1. In the **Explorer**panel, click **Cases**, and then click the case type that you want to open.

**Caution:**Ensure that you open the correct version of your case type if it has multiple parent case types. The instantiation details for a child case type are unique to each parent case type.

1. On the **Settings** tab, click **Instantiation**.
2. Select the **Automatically by system when** check box.
3. Click **Parent case starts**.
4. Optional: Further refine the conditions that control whether a case is created by pressing the Down Arrow key in the **Allow only when** field and selecting the name of a when condition.
5. Click **Save**

# **Configuring dependency instantiation for a case type**

You can configure a child case type with a set of dependencies that control whether the case type is instantiated. By allowing your application to manage run-time dependencies, you can reduce the workload for case workers.

1. In the **Explorer**panel, click **Cases**, and then click the case type that you want to open.

**Caution:**Ensure that you open the correct version of your case type if it has multiple parent case types. The instantiation details for a child case type are unique to each parent case type.

1. On the **Settings** tab, click **Instantiation**.
2. Select the **Automatically by system when** check box.
3. Click [*Any or all*] **of the below conditions are met**.
4. Click **+ Add condition** to define a dependency.
5. In the **Case** field, press the Down Arrow key and select a case type on which your current case type depends.

This list is populated by case types that share your topmost case type and excludes: specialized case types, descendant case types, and case types with more than one parent.

1. In the **Condition** list, select the criteria for creating a child case.
2. If you select the **has work status** condition, select a case status in the list that is displayed.

**Tip:**When using work status as dependency condition, ensure that your process has at least one shape, such as an assignment or utility, that sets the status of a case to the expected value.

1. If you select the **has completed** condition, click **any** or **all** to indicate how many cases must be resolved before a child case is created.
2. Click **OK**.
3. Optional: Further refine the conditions that control whether a case is created by pressing the Down Arrow key in the **Allow only when** field and selecting the name of a when condition.
4. Click **Save**.

At run time, a child case is created if one does not already exist, your dependency conditions are met, and the when condition that you provide returns a true value. The limitation of one child case prevents duplication when a dependency, such as a parent case type reaching a specific stage, is fulfilled more than once.

# **Propagating data to a child case**

You can propagate data from a parent case to each child case that is created at run time. By sharing property values among cases, you save time and provide relevant information to case workers.

1. Open a parent case type in Case Designer.
2. On the **Settings** tab, click **Data propagation** to display a list of child case types and the properties that have been propagated.

For example, **Into Loan (2)** is displayed when two properties are propagated from your case type into the Loan child case type.

1. Click a child case name to expand the section.
2. Click **+ Add property**.
3. In the **Propagate property value** field, press the Down Arrow key and select the name of a property that is copied from the parent case at run time.

**Tip:**Enter a fully qualified name in this field to specify an embedded property. For example, enter .pyWorkParty(Customer).pyWorkPartyUri to set the *pyWorkPartyUri* property that is defined on the Customer page of the *pyWorkParty* page group.

1. In the **To property value** field, press the Down Arrow key and select the name of a property that is set in the child case at run time.
2. Click **OK**.
3. Optional: Select the **Apply data transform** check box and provide the name of a data transform that runs each time that a child case is created.

This data transform does not run when a child case is created by a **Case** step in a stage of the parent case type.

**Tip:**Reference the *pyParentPageName* page property in this data transform to access properties in the parent case type.

1. Repeat steps 3 through 8 to propagate the values of more properties from the parent case type.
2. Click **Save**.

At run time, values are propagated each time that a child case is added to the parent case.