

# Relations Function

V2.0.0

Release Date: 09/2023

## Overview



The Relations function is meant to provide the ability to allow parent/child relations levels within Resilient to link incidents “manually”. This function allows users sync notes from child to parent and vice versa, close all child incidents under a parent, and update the parent when the status of the child changes.

This document outlines the functionality of function as it relates to Resilient.

## Installation

### Integration

#### Required Permissions

Name	Permissions
Org Data	Read
Function	Read
All Incidents	Read
All Incidents Status	Edit
All Incidents Fields	Edit
All Incidents Notes	Edit

To install the function run the following commands to update the Resilient Environment:

- `sudo pip install --upgrade pip`
- `sudo pip install --upgrade setuptools`
- `sudo pip install --upgrade resilient-circuits`

Then run the following command to install the function:

- `sudo pip install --upgrade fn_relations-2.0.0.tar.gz`

To configure this newly installed function run the following commands per the instructions:

#### New environment:

- `resilient-circuits config -c`

### Existing environment:

- `resilient-circuits config -u -l fn-relations`

There is nothing needed in `app.config` document as of version 1.0.0. Therefore, finish by customizing `Resilient` for the relations function.

- `resilient-circuits customize -y -l fn-relations`

Run `resilient-circuits` or restart the service.

- `resilient-circuits run`
- `systemctl restart resilient_circuits`

## AppHost

To install the App: open `Resilient`, navigating to the Administration page, Apps tab and select Install:

- `sudo pip install open Resilient`, navigating to the Administration page and Apps tab
- select Install, uploading the zip file for `fn_relations`, and following the steps to install
- Choose the AppHost server to deploy the app
- Select Deploy

## Release Notes

Version	Date	Notes
1.0.0	07/2020	Initial Release
1.0.1	05/2021	Add AppHost Support
1.0.2	05/2023	Support for Python 3.9 Support for CP4S Patch: Verification of Parent and Child Incidents are different Patch: Changed rules to only sync incident notes.
2.0.0	09/2023	New Function: Copy Task to Children New Function: Sync Task Notes from Copied Tasks New Function: Sync Artifact Data to Parent and Children New Function: Sync DataTable Data to Parent and Children Enhancement: Added functionality to all note syncing to allow conversations, meaning notes and replies to notes for better collaboration

## Design Changes

The design that is recommended to layout the new fields and data tables is following:

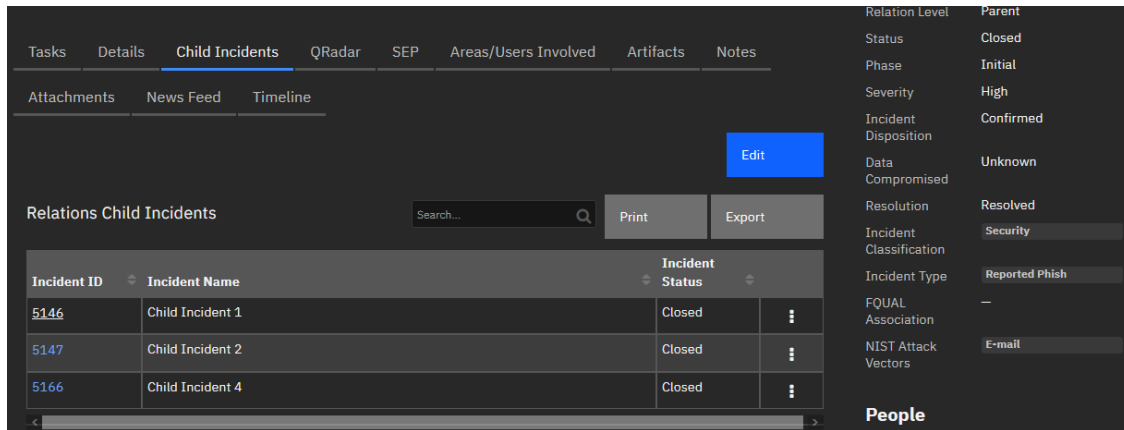
### If Relation Level is: Parent

New Tab: Child Incidents

- Add Relations Child Incidents data table

Summary Section:

- Add Relation Level



### If Relation Level is: Child

#### Summary Section:

- Add Relation Level
- Parent ID

Summary	
ID	5146
Relation Level	Child
Parent ID	5169
Status	Closed
Phase	Initial
Severity	Info

## Function Descriptions

Once the function has been deployed, you can access use the functions within Resilient. The available functions are pictured below.

Name	Description	
Relations: Assign Parent	Create a parent/child relationship between the 2 incidents provided.	
Relations: Auto Close Child Incidents	Close child incidents when the parent incident is closed.	
Relations: Copy Task	Copy a task from a Parent Incident down to the Children.	
Relations: Remove Child Relation	Used to remove the relation child relation from a Child incident as well as removing the parent relation from the Parent incident if it no longer has children.	
Relations: Sync Artifact	Sync Artifacts from the incident where the artifact is currently to the parent or child.	
Relations: Sync Child Table Data	Update data within the Parent Table if the Child data changes.	
Relations: Sync DataTable Data	A Function used to Sync DataTable Data from the incident where it resides to the parent or child.	
Relations: Sync Notes	Sync notes from the incident where the note is currently to the parent or child.	
Relations: Sync Task Notes	Sync Task notes from a copied Task back to parent originating Task.	

## Workflow Descriptions

The example workflows are meant to be used as a guide to understand how to properly use the function. The pre-process scripts are designed to be used out of the box and there is no need for post-process scripts as there will be no need to data manipulation after the function is complete.

Workflow Name	Description	Object Type	Rules	
Example: Relations: Assign Parent	Change the necessary information to establish a child/parent relationship.	Incident	Example: Relations - Assign Parent Incident	
Example: Relations: Auto Close Child Incidents	Close the incidents of the child incidents when the parent incident is closed.	Incident	Example: Relations - Close Child Incidents	
Example: Relations: Remove Child Relation	Removes the child incident relation with the parent.	Incident	Example: Relations - Remove Child Relation	
Example: Relations: Send Task to Children	Copy a task from a parent to the children incidents.	Task	Example: Relations - Send Task to Children	
Example: Relations: Sync Artifact to Parent/Child	Sync any new notes created in the parent or child to either the children incidents (if this is labeled as a parent) or the parent incident (if this is labeled as a child).	Artifact	Example: Relations - Sync Artifact	
Example: Relations: Sync DataTable Data to Parent/Child	Sync Data from DataTables to the incidents Parent or Child to track data across incidents.	Incident	Example: Relations - Sync DataTable Data	
Example: Relations: Sync Notes to Parent/Child	Sync any new notes created in the parent or child to either the children incidents (if this is labeled as a parent) or the parent incident (if this is labeled as a child).	Note	Example: Relations - Sync Notes with Child Example: Relations - Sync Notes with Parent	
Example: Relations: Sync Task Notes to Parent/Child	Sync Task Notes to Parent/Children Task(s).	Note	Example: Relations - Sync Task Notes to Parent Example: Relations - Sync Task Note to Child	
Example: Relations: Update Child Table Data	Update any data stored in the Child Incidents Data Table on the parent incident if changed, such as if the incident is closed.	Incident	Example: Relations - Update Child Incident Parent Data Table	

## Rule Descriptions

The rules are designed to be a guide as to how to setup your rules. They are setup as to only appear on simulation only to cut back on the clutter brought in on live incidents while testing and tweaking this function. If you want to make the example rules available, just remove the simulation requirement.

Order	Rule Name	Process Type	Object Type	Conditions	Enabled
19	Example: Relations - Close Child Incidents	Automatic	Incident	Date Closed Simulation Relation Level	
20	Example: Relations - Sync Notes with Parent	Automatic	Note	Simulation Relation Level Parent ID Text ID Note is created	
21	Example: Relations - Update Child Incident Parent Data Table	Automatic	Incident	Simulation Name Status Relation Level Parent ID	
23	Example: Relations - Sync Task Notes to Parent	Automatic	Note	Simulation Relation Level Text ID Name Note is created	
-	Example: Relations - Assign Parent Incident	Menu Item	Incident	Simulation Relation Level	
-	Example: Relations - Remove Child Relation	Menu Item	Incident	Simulation Relation Level	
-	Example: Relations - Send Task to Children	Menu Item	Task	Simulation Relation Level	
-	Example: Relations - Sync Artifact	Menu Item	Artifact	Simulation Relation Level	
-	Example: Relations - Sync DataTable Data	Menu Item	Incident	Simulation Relation Level	
-	Example: Relations - Sync Notes with Child	Menu Item	Note	Simulation Relation Level ID	
-	Example: Relations - Sync Task Note to Child	Menu Item	Note	Simulation Relation Level ID	

## Using the Function

Adding a Child to a parent.

Requirements:

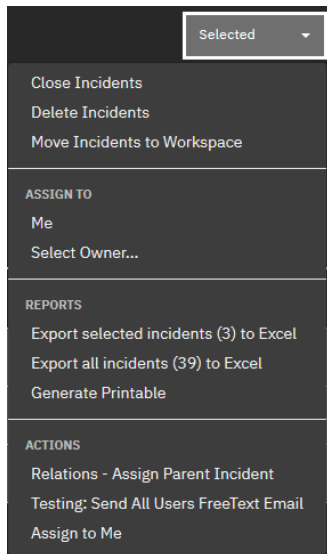
Parent Incident must already be created and user need to know Parent Incident ID. And the incidents you are going to be turning into a child incident can't have an active relation.

Instructions:

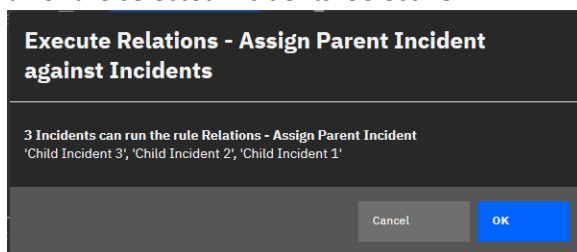
1. Either select the incident in the Incidents View of Resilient pictured below or go to the next step if you are already in a single incident.

<input type="checkbox"/>	ID	Severity	Name	Description
<input type="checkbox"/>	5101	Info	<span>SIM</span> Parent Incident	—
<input checked="" type="checkbox"/>	5100	Info	<span>SIM</span> Child Incident 3	—
<input checked="" type="checkbox"/>	5099	Info	<span>SIM</span> Child Incident 2	—
<input checked="" type="checkbox"/>	5098	Info	<span>SIM</span> Child Incident 1	—

2. Select the Menu in the top right and select the “Example: Relations – Assign Parent Incident”.



3. If multiple incidents were selected, it may ask if you are sure you want to apply it to all of the selected incidents. Select “OK”.



4. Enter in the Parent Incident ID



Results: The following things will be updated.

Child Incident:

Incident fields Relation Level is set to Child and Parent ID is added as a clickable field to open a new tab of the parent incident located in the summary section to the right within the incident.

Summary	
ID	5100
Relation Level	Child
Parent ID	<a href="#">5101</a>

Artifact Related Parent Incident is added with the parent ID to correlate the incidents within the application.

Hits	Related Incidents	Type	Value
3		Related Parent Incident	5101

Parent Incident: \*If this is not already a parent.

Incident field Relation Level is set to Parent located in the summary section to the right within the incident.

Summary	
ID	5101
Relation Level	Parent

The same artifact above is created.

The incident is added to the Relations Child Incidents data table located in the new tab Child Incidents (the Incident ID of each incident is clickable).

Relations Child Incidents			
Search...		Print	Export
Incident ID	Incident Name	Incident Status	
5100	Child Incident 3	Closed	
5098	Child Incident 1	Closed	
5099	Child Incident 2	Closed	
Displaying 1 - 3 of 3			

Any notes on the child are synced to the parent with the top 3 lines of the following note added to the beginning of any synced notes followed by the actual note.

Parent/Child Relationships added a note to the Incident 08/31/2023 00:39

Note from Child Incident: 2804  
Note ID: 615  
On Date: 08/31/2023 03:57:05  
By: John Doe

Note from Child Pre Attachment!

Removing a Child from a Parent:

Requirements:

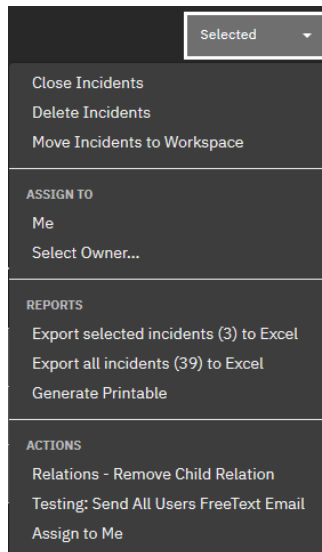
Incident must have a Relation Level of Child already.

## Instructions:

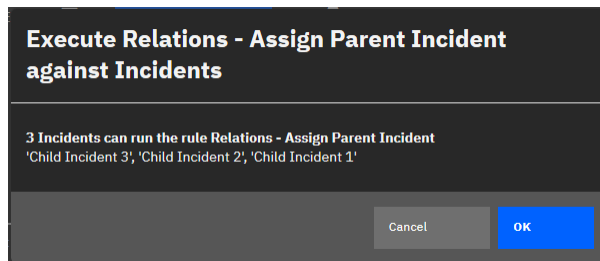
1. Either select the incident in the Incidents View of Resilient pictured below or go to the next step if you are already in a single incident.

<input type="checkbox"/>	ID	Severity	Name	Description
<input type="checkbox"/>	5101	Info	<span>SIM</span> Parent Incident	—
<input checked="" type="checkbox"/>	5100	Info	<span>SIM</span> Child Incident 3	—
<input checked="" type="checkbox"/>	5099	Info	<span>SIM</span> Child Incident 2	—
<input checked="" type="checkbox"/>	5098	Info	<span>SIM</span> Child Incident 1	—

2. Select the Menu in the top right and select the “Example: Relations – Remove Child Relation”.



3. If multiple incidents were selected, it may ask if you are sure you want to apply it to all of the selected incidents. Select “OK”.





4. Choose if you want to remove the synced notes between this relationship.



Results: The following things will be removed.

Child Incident:

Fields Relation Level and Parent ID values will be removed.

Artifact Related Parent Incident will be removed.

If you selected to remove Notes, Any notes synced down from the parent will also be removed.

Parent Incident:

The child will be removed from the Relations Child Incidents data table.

If you selected to remove Notes, Any notes synced from the child to the parent will also be removed.

If that was the last child of the parent, the Artifact field Relation level, artifact Related Parent Incident, and Child Incidents Tab will be removed.

Syncing Notes from Child to Parent:

Requirements:

The incident must have a Relation Level of Child.

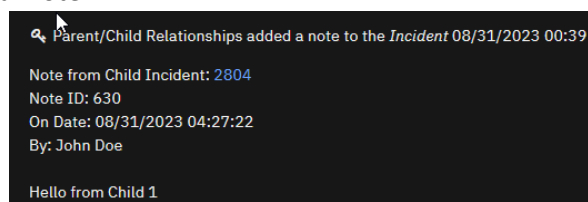
Instructions:

1. All notes from the child are automatically synced with the parent.

Results:

Parent Incident:

Any notes on the child are synced to the parent with the top 3 lines of the following note added to the beginning of any synced notes followed by the actual note.



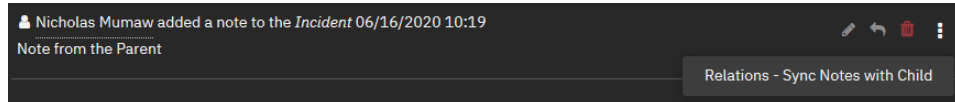
## Syncing Notes from Parent to Child:

### Requirements:

The incident must have a Relation Level of Parent. The note must already have been added to the parent incident.

### Instructions:

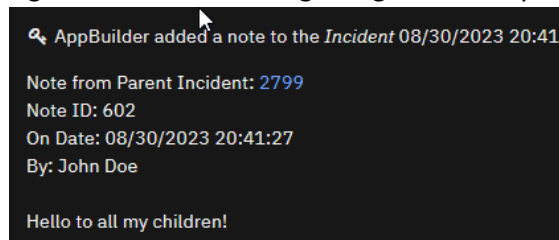
1. Within the parent Note tab, click the vertical ellipsis on the note you want to sync and select the “Example: Relations – Sync Notes with Child”.



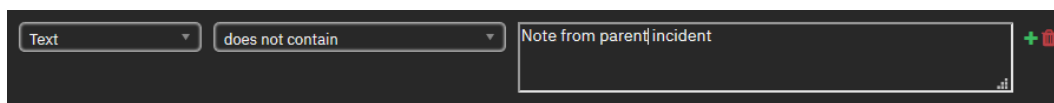
### Results:

#### Children Incident:

Any note synced will sync with all the children with the top 3 lines of the following note added to the beginning followed by the actual note.



Notes:\*\*\* To prevent synced parent notes to be synced back to the parent put this condition into the rule.



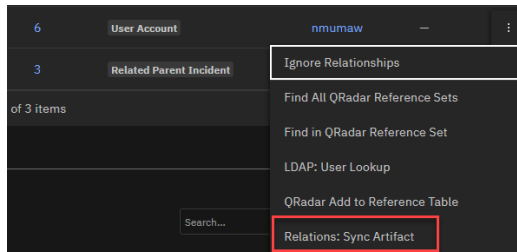
## Syncing Artifacts to the Parent or Child:

### Requirements:

The incident must have a Relation Level.

### Instructions:

1. Within the Artifact tab of any Parent or Child, click the vertical ellipsis on the artifact you want to sync and select the “Example: Relations – Sync Artifact”.



#### Results:

##### Parent Incident:

Will now have the Artifact in its list of artifacts.

##### Child Incidents:

All children will not have the parent artifact in their artifacts.

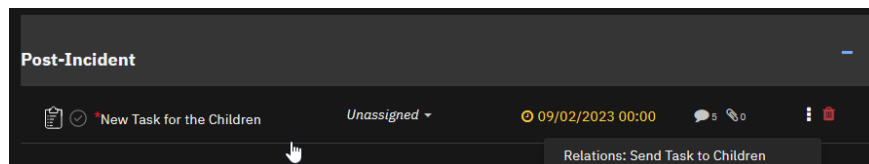
#### Copying a Parent Task to a Child:

##### Requirements:

The incident must have a Relation Level of Parent.

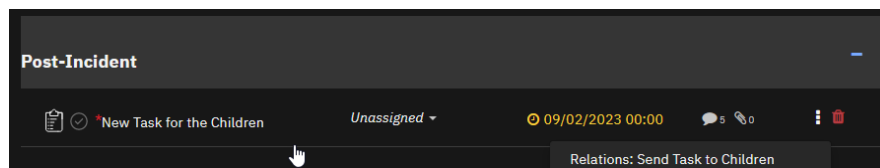
##### Instructions:

Click the vertical ellipsis on the task you want to copy then select “Example: Relations - Send Task to Children”.



#### Results:

All of the children will have the Task that was selected from the parent with the same due date, instructions, and name as below.



#### Syncing a DataTable with a parent or child:

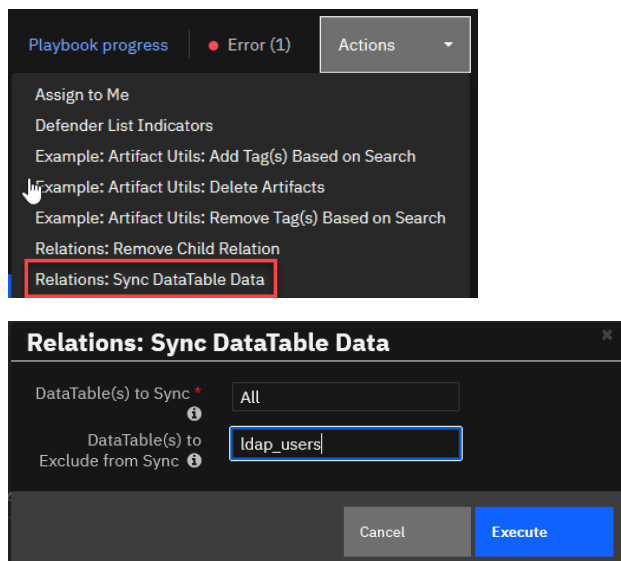
##### Requirements:

The incident must have a Relation Level.

##### Instructions:

In the top right of the screen click on “Actions” then select “Example: Relations - Sync

DataTable Data". You will receive a pop-up menu asking what DataTables you want to sync, either enter a comma separated list of the API names of the DataTables you want to sync or type "All" to sync all of the DataTables. If selecting "All", but you would like to exclude any, just provide a comma separated list of DataTables to ignore. Finally Click Execute.



#### Results:

All of the DataTables requested will have all of their contents synced over to either the parent or children incidents, unless the data already exists, then it will not add the rows already there.

**\*\*Note:** This function also has the capability to sync individual rows if the user wants to build the playbook for that.

#### Updating the child details in the Data Table of the Parent:

##### Requirements:

The incident must have a Relation Level of Child.

##### Instructions:

1. If the Child Incident Name or Status is changed, the data table will be automatically updated.

#### Results:

##### Parent Incident:

The child will be updated in the Relations Child Incident data table.

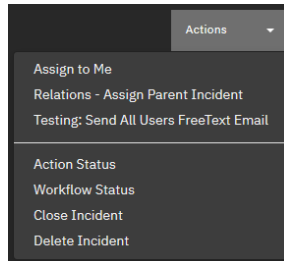
Closing all child incidents from the Parent:

Requirements:

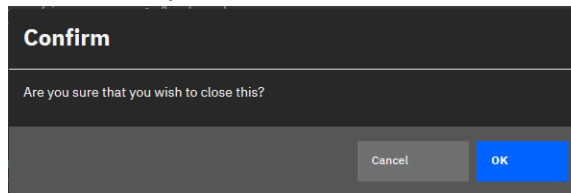
The incident must have a Relation Level of Parent.

Instructions:

1. Select the Menu in the top right and select the “Close Incident”.



2. Confirm that you want to close this incident.



3. Fill out any fields required/desired and select “OK”.

Results:

Parent Incident:

The parent incident will be closed with the fields updated as you filled out in the form.

Child Incident:

The children will also be closed with the same field updates as the parent.

## Design of the Overall Function

Field(s) to Create:

Relation Level: Select

Purpose: Determine the relation to other incidents.

Options:

None

Parent

Child

Parent ID: Text Area (rich text)

Purpose: Show the Incident ID of the parent incident that is clickable to the parent case.

Parent Incident:

Relation Level	Parent
----------------	--------

Child Incident:

Relation Level	Child
Parent ID	5076

Data Table(s) to Create:

Data Table Name: Relations Child Incidents

Rows:

Incident ID

Note: Clickable to the child case

Incident Name

Incident Status

Parent Incident:

Relations Child Incidents			
Incident ID	Incident Name	Incident Status	
5053	Testing Child Incident	Closed	⋮
5055	Testing Child Case 2	Closed	⋮

Artifact(s) to Create:

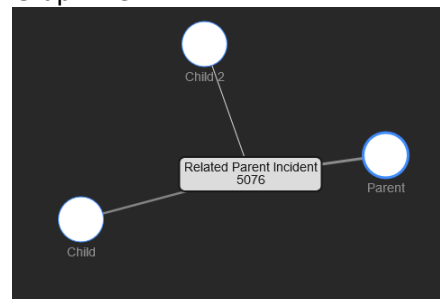
Related Parent Incident

Purpose: Incident ID of the parent of all related incidents to create a relation within Resilient incidents manually.

Table View:

Hits	Related Incidents	Type	Value	Created
2		Related Parent Incident	5076	06/02/2020 16:35

Graph View:



Message Destination(s) to Create:

fn\_relations

Function(s) to Create:

Relations: Assign Parent

Purpose: Create a parent/child relationship between the 2 incidents provided.

Inputs: (\* designates required)

relations\_child\_incident\_id\*: Number

relations\_parent\_incident\_id\*: Number

Relations: Sync Notes

Purpose: Sync notes from the incident where the note is currently to the parent or child.

Inputs: (\* designates required)

relations\_note\_id\*: Number

incident\_id\*: Number

Relations: Sync Child Table Data

Purpose: Update data within the Parent Table if the Child data changes.

Inputs: (\* designates required)

relations\_child\_incident\_id\*: Number

relations\_parent\_incident\_id\*: Number

Relations: Auto Close Child Incidents

Purpose: Close child incidents when the parent incident is closed.

Inputs: (\*designates required)

relations\_parent\_incident\_id\*: Number

Relations: Remove Child Relation

Purpose: Used to remove the child relations from a child incident as well as the parent relation if it is no longer a parent.

Inputs: (\*designates required)

Relations\_child\_incident\_id\*: Number

Relations\_remove\_notes: Boolean

#### Workflow(s) to Create:

##### Example: Relations: Assign Parent

Purpose: Change the necessary information to establish a child/parent relationship.

Workflow Type: Incident

##### Example: Relations: Sync Notes to Parent/Child

Purpose: Sync any new notes created in the parent or child to either the children incidents (if this is labeled as a parent) or the parent incident (if this is labeled as a child).

Workflow Type: Note

##### Example: Relations: Update Child Table Data

Purpose: Update any data stored in the Child Incidents Data Table on the parent incident if changed, such as if the incident is closed.

Workflow Type: Incident

##### Example: Relations: Auto Close Child Incidents

Purpose: Close the incidents of the child incidents when the parent incident is closed.

Workflow Type: Incident

##### Example: Relations: Remove Child Relation

Purpose: Change the necessary information to remove the established child/parent relationship.

Workflow Type: Incident

#### Rule(s) to Create:

##### Example: Relations - Assign Parent Incident

Rule Deployment: Menu

Purpose: Allow the user to select which incident should be set as the parent incident and kick off the workflow to perform the necessary changes.

Rule Type: Incident

Required Fields from user:



Parent Incident: Incident ID of the parent

Workflow to Kickoff: Example: Relations: Assign Parent

Note(s): \*\*Incident will need to not be labeled as a child or parent already to ensure that it does not interfere with already established relationships.

Example: Relations – Sync Notes with Parent

Rule Deployment: Automatic

Purpose: Automatically sync any new notes made on a child incident with a parent incident.

Rule Type: Note

Workflow to Kickoff: Example: Relations: Sync Notes to Parent/Child

Example: Relations – Sync Notes with Child

Rule Deployment: Menu

Purpose: Allow a note from a parent incident to manually be synced down to the child incidents.

Rule Type: Note

Workflow to Kickoff: Example: Relations: Sync Notes to Parent/Child

Notes(s): None

Example: Relations – Update Child Incident Parent Data Table

Rule Deployment: Automatic

Purpose: Any changes in the data in the data table if changed will get automatically updated.

Rule Type: Incident

Workflow to Kickoff: Example Relations: Update Child Table Data

Note(s): \*\*If the incident is a child incident and has a parent id listed and if either the name or status of the incident is changed, this will kick off.

Example: Relations - Close Child Incidents

Rule Deployment: Automatic

Purpose: If the parent incident is closed then close the child incidents automatically.

Rule Type: Incident

Workflow to Kickoff: Example: Relations: Auto Close Child Incidents

Note(s): \*\*If the parent incident date closed has value, then it will close all child incidents.

Example: Relations - Remove Child Relation

Rule Deployment: Menu

Purpose: Allow the user to remove a child incidents parent relation to kick off the workflow to perform the necessary changes.

Rule Type: Incident

Required Fields from user:

Remove Notes from Relation?: Boolean to give the user the option to remove synced notes.

Workflow to Kickoff: Example: Relations: Assign Parent

Note(s): \*\*Incident will need to be labeled as a child already.

## Troubleshooting

Please feel free to reach out to me on the IBM Community or make a post in the community for any support.