

---

# Apply changes

## Definition

This step is used to apply the identity changes to the Soffid repository.

## Steps Tabs

### Task details

- **Apply users changes:** check it (select the Yes option) to make changes to users on the Soffid repository.
- **Apply entitlements:** check it (select the Yes option) to make changes to permissions on the Soffid repository.

## Incoming transitions

The Incoming transitions tab displays the previous steps where the flow comes from. When you create a process from a template or from scratch default incoming transitions are defined. It is allowed to customize the default setup, add new transitions, or delete transitions.

- **From:** the previous step, where the flow comes. Allows you to select where the workflow comes from.
- **Incoming transition:** brief name to identify the transition. That is the name of the action the form will show to the final user.
- **To:** current step.
- **Action:** allows creating a custom script to perform specific actions.

When you create an incoming transition, Soffid creates the proper outgoing transition.

## Example

Scroll through the list of values to perform some operations.

```
grants = executionContext.getVariable("grants");
for (roleRequestInfo:grants) {
    // TO-DO
}
```

## Outgoing transitions

The Outcoming transition tab displays the next steps where the flow can go from the current step. When you create a process from a template or from scratch default outcoming transitions are defined. It is allowed to customize the default setup, add new transitions, or delete transitions.

- **From:** current step.
- **Incoming transition:** name of the transition.
- **To:** the next step, where the flow goes.
- **Action:** allows creating a custom script to perform specific actions.

When you create an outcoming transition, Soffid creates the proper incoming transition.

### Example

If the user's country is Spain, it will delete all the groups to which the user belongs:

```
userName = executionContext.getVariable("userName");
user = serviceLocator.getUserService().findUserByUserName(userName);
country = user.getAttributes().get("country");
groups = serviceLocator.getGroupService().findUsersGroupByUserName(userName);

if (country.equals("ES")) {
    for (groupUser: groups) {
        serviceLocator.getGroupService().removeGroupFormUser(userName, groupUser.group);
    }
}
```