

1. OnStart Property

```
/*  
Noticed that your crew type datasource is providing just the  
members, and the Admins in another one. With a small change we can  
improve that and condense in just one list  
Furthermore, you are using a distinct function (not delegable) to  
get the Crews, it's better to just create a new list of CrewType  
with just the name and an ID, relating the two lists afte  
In this app, there are just collections to make It work  
*/
```

```
//variable to store the User informations
```

```
Set(  
    varCurrentUser;  
    User()  
);;
```

```
//CrewTypeNames list (new list to store all the crewTypes you  
have), using GUID to generate a new rowID - don't relying on  
sharepoint's ID)
```

```
//The new list contains the admin too. I recommend to store this  
list in a collection, because I believe it would be a small one
```

```
ClearCollect(  
    colCrewTypeNames;  
    {  
        id: Text(GUID());  
        name: "Crew1"  
    };  
    {  
        id: Text(GUID());  
        name: "Crew2"  
    };  
    {  
        id: Text(GUID());  
        name: "Admin"  
    }  
);;
```

```
//CrewType list, but instead of using the name, relating the  
members to ID, avoiding the use of Distinct later
```

```
//This one is a sharepoint simulation
```

```
ClearCollect(  
    colCrewType;  
    {  
        member: varCurrentUser;  
        id: Index(colCrewTypeNames;3).id  
    };  
);;
```

```

    {
        member: varCurrentUser;
        id: Index(colCrewTypeNames;2).id
    };
    {
        member: varCurrentUser;
        id: Index(colCrewTypeNames;3).id
    }
};

```

//Setting a new variable to store the user's role and IsAdmin value (Record), changing filter to lookup to improve performance

```

Set(
    varCurrentUserExtra;
    With(
        {
            _1:
                //with acts like a variable for just this formula
                Lookup(
                    colCrewType;
                    member.Email = varCurrentUser.Email
                )
        };
        With(
            {
                _12:
                    Lookup(
                        colCrewTypeNames;
                        id = _1.id
                    )
            };
            {
                crew:_12;
                isAdmin: _12.name = "Admin"
            }
        )
    )
)

```

```

);;
//This collection is used just to set de default value of the
column "items" of colSelected and avoid some posterior errors
ClearCollect(
    colClear;
    ""
);;

```

//This collection used to store all options when someone uses the dropdowns. We need this approach to avoid circular references on app

//Instead of an unique collection, it's possibel to create one to each combobox

```
ClearCollect(
    colSelected;
    {
        cmbName: "cmbAssetClass";
        items: Clear(colClear)
    };
    {
        cmbName: "cmbCompliance";
        items: Clear(colClear)
    };
    {
        cmbName: "cmbMaintType";
        items: Clear(colClear)
    };
    {
        cmbName: "cmbSubstation";
        items: Clear(colClear)
    };
    {
        cmbName: "cmbAssetLocation";
        items: Clear(colClear)
    };
    {
        cmbName: "cmbAssetType";
        items: Clear(colClear)
    };
    {
        cmbName: "cmbCrewTypeNames";
        items:
        If(
            varCurrentUserExtra.crew.name = "Admin";
            Clear(colClear);
            {Value: varCurrentUserExtra.crew.name}
        )
    }
);;
```

//Here there is a simulation of your data source, just a Note, I created a collection to each one of the columns to make it easier

```
Concurrent(
    ClearCollect(
        colAssetClass;
```

```
{
    id: Text(GUID());
    name: "AssetClass1"
};
{
    id: Text(GUID());
    name: "AssetClass2"
};
{
    id: Text(GUID());
    name: "AssetClass3"
}
);
ClearCollect(
    colAssetType;
    {
        id: Text(GUID());
        name: "colAssetType1"
    };
    {
        id: Text(GUID());
        name: "colAssetType2"
    };
    {
        id: Text(GUID());
        name: "colAssetType3"
    }
);
ClearCollect(
    colAssetLocation;
    {
        id: Text(GUID());
        name: "AssetLocation1"
    };
    {
        id: Text(GUID());
        name: "AssetLocation2"
    };
    {
        id: Text(GUID());
        name: "AssetLocation3"
    }
);
ClearCollect(
    colSubstation;
    {
        id: Text(GUID());
        name: "Substation1"
    };
};
```

```

        {
            id: Text(GUID());
            name: "Substation2"
        };
        {
            id: Text(GUID());
            name: "Substation3"
        }
    );
    ClearCollect(
        colMaintenanceTaskType;
        {
            id: Text(GUID());
            name: "MaintenanceTaskType1"
        };
        {
            id: Text(GUID());
            name: "MaintenanceTaskType2"
        };
        {
            id: Text(GUID());
            name: "MaintenanceTaskType3"
        }
    );
    ClearCollect(
        colCompliance;
        {
            id: Text(GUID());
            name: "Compliance1"
        };
        {
            id: Text(GUID());
            name: "Compliance2"
        };
        {
            id: Text(GUID());
            name: "Compliance3"
        }
    )
);

//Your main datasource with example items
//This one is a sharepoint simulation
ClearCollect(
    colScheduledMaintenance;
    {
        Crew: Index(colCrewTypeNames;1).name;
        'Asset Class': Index(colAssetClass;2).name;
    }
);

```

```

    'Asset Type': Index(colAssetType;1).name;
    'Asset Location': Index(colAssetLocation;3).name;
    Substation: Index(colSubstation;2).name;
    'Maintenance Task Type':
Index(colMaintenanceTaskType;1).name;
    Compliance: Index(colCompliance;2).name
};
{
    Crew: Index(colCrewTypeNames;1).name;
    'Asset Class': Index(colAssetClass;3).name;
    'Asset Type': Index(colAssetType;2).name;
    'Asset Location': Index(colAssetLocation;2).name;
    Substation: Index(colSubstation;1).name;
    'Maintenance Task Type':
Index(colMaintenanceTaskType;2).name;
    Compliance: Index(colCompliance;1).name
};
{
    Crew: Index(colCrewTypeNames;1).name;
    'Asset Class': Index(colAssetClass;3).name;
    'Asset Type': Index(colAssetType;3).name;
    'Asset Location': Index(colAssetLocation;3).name;
    Substation: Index(colSubstation;3).name;
    'Maintenance Task Type':
Index(colMaintenanceTaskType;3).name;
    Compliance: Index(colCompliance;3).name
}
)

```

Name of the inputs

2. Combobox cmbCrewTypeNames Property

The screenshot shows a web application interface with a filter form and a data table. The filter form includes dropdowns for cmbCrewTypeNames, cmbAssetClass, cmbAssetType, cmbAssetLocation, cmbSubstation, cmbMaintType, and cmbCompliance. A 'clearFilters' button is also present. Below the form is a table with columns: Crew, Asset Class, Asset Type, Asset Location, Substation, Maintenance Task Type, and Compliance. The table contains three rows of data.

| Crew | Asset Class | Asset Type | Asset Location | Substation | Maintenance Task Type | Compliance |
|-------|-------------|---------------|----------------|-------------|-----------------------|-------------|
| Crew1 | AssetClass2 | colAssetType1 | AssetLocation3 | Substation2 | MaintenanceTaskType1 | Compliance2 |
| Crew1 | AssetClass3 | colAssetType2 | AssetLocation2 | Substation1 | MaintenanceTaskType2 | Compliance1 |
| Crew1 | AssetClass3 | colAssetType3 | AssetLocation3 | Substation3 | MaintenanceTaskType3 | Compliance3 |

2.1 Items

```
If(
    !varCurrentUserExtra.isAdmin;
    Table(
        {
            Value: varCurrentUserExtra.crew.name
        }
    );
    Distinct(
        galScheduledMaintenance.AllItems.Crew;
        ThisRecord.Crew
    )
)
```

2.2 On Change

```
// on OnChange, This input patch de colSelected to
Self.SelectedItems, what filters the gallery
```

```
Patch(
    colSelected;
    LookUp(colSelected; cmbName = "cmbAssetClass");
    {
        items: Self.SelectedItems
    }
)
```

2.3 DisplayMode

```
//If the current user isn't an Admin, the input is disabled
If(
    varCurrentUserExtra.isAdmin;
    DisplayMode.Edit;
    DisplayMode.Disabled
)
```

2.4 Reset

```
cxtReset
```

2.5 DefaultSelectedItems

```
If(
    !varCurrentUserExtra.isAdmin;
    {Value: varCurrentUserExtra.crew.name}
)
```

3. Other Comboboxes Properties

3.1 Items

```
Distinct(  
    galScheduledMaintenance.AllItems.'Asset Class';  
    ThisRecord.'Asset Class'  
)
```

3.2 OnChange

```
Patch(  
    colSelected;  
    LookUp(colSelected; cmbName = "cmbAssetClass");  
    {  
        items: Self.SelectedItems  
    }  
)
```

3.3 Reset

cxtReset

4. Clear filters Property

The screenshot shows a web application interface with a form containing several dropdown menus and a 'clearFilters' button. The form fields are:

- cmbCrewTypeNames
- cmbAssetClass
- cmbAssetType
- cmbAssetLocation
- cmbSubstation
- cmbMaintType
- cmbCompliance

A red box highlights the 'clearFilters' button. Below the form, there is a table with the following columns: Crew, Asset Class, Asset Type, Asset Location, Substation, Maintenance Task Type, and Compliance. The table contains three rows of data:

| Crew | Asset Class | Asset Type | Asset Location | Substation | Maintenance Task Type | Compliance |
|-------|-------------|---------------|----------------|-------------|-----------------------|-------------|
| Crew1 | AssetClass2 | colAssetType1 | AssetLocation3 | Substation2 | MaintenanceTaskType1 | Compliance2 |
| Crew1 | AssetClass3 | colAssetType2 | AssetLocation2 | Substation1 | MaintenanceTaskType2 | Compliance1 |
| Crew1 | AssetClass3 | colAssetType3 | AssetLocation3 | Substation3 | MaintenanceTaskType3 | Compliance3 |

4.1 OnSelect

```
UpdateContext(  
    {  
        cxtReset: true  
    }  
);;  
UpdateContext(  
    {  
        cxtReset: true  
    }  
);;
```



```

    {
      cxtReset: false
    }
  );;

```

5. Gallery property

cmbCrewTypeNames

cmbAssetClass

cmbAssetType

clearFilters

cmbAssetLocation

cmbSubstation

cmbMaintType

cmbCompliance

THERE ARE DELEGATION WARNINGS USING THIS APPROACH

| Crew | Asset Class | Asset Type | Asset Location | Substation | Maintenance Task Type | Compliance |
|-------|-------------|---------------|----------------|-------------|-----------------------|-------------|
| Crew1 | AssetClass2 | colAssetType1 | AssetLocation3 | Substation2 | MaintenanceTaskType1 | Compliance2 |
| Crew1 | AssetClass3 | colAssetType2 | AssetLocation2 | Substation1 | MaintenanceTaskType2 | Compliance1 |
| Crew1 | AssetClass3 | colAssetType3 | AssetLocation3 | Substation3 | MaintenanceTaskType3 | Compliance3 |

5.1 Items

```

/*
INDEX TABLE
{
1   cmbName: "cmbAssetClass";
    items: Clear(colClear)
};
{
2   cmbName: "cmbCompliance";
    items: Clear(colClear)
};
{
3   cmbName: "cmbMaintType";
    items: Clear(colClear)
};
{
4   cmbName: "cmbSubstation";
    items: Clear(colClear)
};
{
5   cmbName: "cmbAssetLocation";
    items: Clear(colClear)
};
};

```

```

    {
    6     cmbName: "cmbAssetType";
        items: Clear(colClear)
    };
    {
    7     cmbName: "cmbCrewTypeNames";
        items: Clear(colClear)
    }
}

*/
//Note: I'm using index here to grab the data from the collection
(the performance is better than LookUp), but you can create
different collections for the comboboxes, you decide which is
better
Filter(
    colScheduledMaintenance As Tr;
    //The first filter is to make sure that the gallery is
filtering the items no Admins ad other in the right way
    Or(
        //It's necessary to use this formula:
        "IsBlank(First(Index(colSelected;7).items).Value)", because when a
        combobox resets, It run onChange and patch colSelected with a non
        blank array, there are other ways to avoid this problema
        varCurrentUserExtra.isAdmin And
        IsBlank(First(Index(colSelected;7).items).Value);
        //Note: I adapted your formula to "in" instead, because the
        final user can select multiple items to filter de collection
        //The crew columns is defferent due to the validation of
        admin, then the cmbCrewTypeNames is disabled too
        Tr.Crew in Index(colSelected;7).items
    ) And
    And(
        Tr.'Asset Class' in Index(colSelected;1).items ||
        IsBlank(First(Index(colSelected;1).items));
        Tr.Compliance in Index(colSelected;2).items ||
        IsBlank(First(Index(colSelected;2).items));
        Tr.'Maintenance Task Type' in Index(colSelected;3).items ||
        IsBlank(First(Index(colSelected;3).items));
        Tr.Substation in Index(colSelected;4).items ||
        IsBlank(First(Index(colSelected;4).items));
        Tr.'Asset Location' in Index(colSelected;5).items ||
        IsBlank(First(Index(colSelected;5).items));
        Tr.'Asset Type' in Index(colSelected;6).items ||
        IsBlank(First(Index(colSelected;6).items))
    )
)
)

```