

# 1 Group rights

On page “Settings> Group rights” you can configure the rights for the user groups.

The OP modes are displayed in the header of the table:






Common   Color Theme   Login   User Manager   Group rights				
User Group	operator ▼	Load default	Save config	
	Item name	iOpSetup	iOpManual	iOpAuto
–	iMmGeneral			
	...iOptOverride			
	...iOptMachineName			
	...iOptHideMenues			
	...iOptHelpButton			
	...iOptPosInHeader			
+	iMmProduction			
+	iMmRecipe			
+	iMmDiag			
+	iMmCommisioning			
+	iMmSettings			
+	iMmDebug			

You can select the following options for the OP mode buttons:

Icon	Meaning	Virtual constant item
	button is visible and enabled	.EDH_EDIT=0
	button is visible but locked	.EDH_LOCKED=3
	button is invisible	.EDH_HIDDEN=4

The main menu items, have the same options as the OP mode buttons.

The sub menu items, have the following options:

Icon	Meaning	Virtual constant item
	Sub menu is selectable. Content of sub menu screen is editable by multiple clients	.EDH_EDIT=0
	Not yet implemented. Sub menu is selectable. Content of sub menu screen is editable by a single client. The first client which open the screen can edit it. The screen is locked until the client closes the screen	.EDH_EDIT_ONCE=0
	Sub menu is selectable. Content of sub menu screen is read only.	.EDH_VIEW=2
	Sub menu is locked	.EDH_LOCKED=3
	Sub menu is invisible	.EDH_HIDDEN=4

The 3<sup>rd</sup> level of group rights allows to set specific features on a sub menu content page.  
E.g. iMmRecipe/iSmTeachPoints/iRecipeOpen. The same options as for sub menu items are available.

## 1.1 Implementation details

In WebIQ code editor open the local script cfg.js, to edit the variables below:

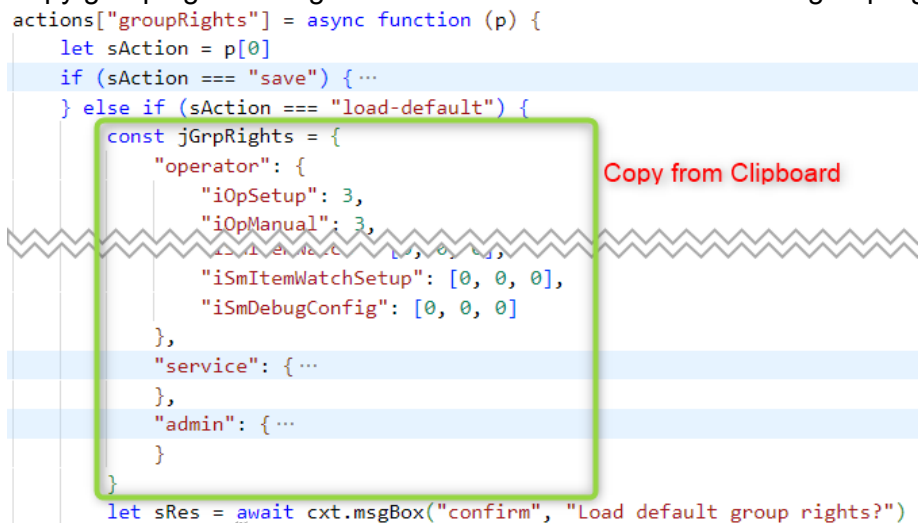
Action	variable
Define user groups	<pre>const arUserGroups = ["operator", "service", "admin"] // User group are displayed in a cx-select widget</pre>
Define OP modes	<pre>const arOpModes = ["iOpSetup", "iOpManual", "iOpAuto"] // Op modes are displayed in header of table</pre>
Define rights	<pre>const arGrpData = [   :   ['iMmRecipe', 'gtl-menuMain', 1], // virtual-item, text-list, level   ['iSmTeachPoints', 'gtl-menuSub', 2], // virtual-item: name of virtual item   ['iRecipeOpen', 'gtl-menuSub', 3], // text-list: name of text list   ['iRecipeEdit', 'gtl-menuSub', 3], // see local script global-textlists   ['iSmSetup', 'gtl-menuSub', 2], // level: level of group right   ['iSmSafeZones', 'gtl-menuSub', 2], // 1=main menu   ['iSmSwitches', 'gtl-menuSub', 2], // 2=sub menu   ['iSmVisualProg', 'gtl-menuSub', 2], // 3=features   : ]</pre>

- Each right is displayed in a single table row
- Each virtual item, which is defined in arGrpData[] must be created with a dot as prefix.  
E.g. iMmRecipe => virtual:iMmRecipe
- Until the virtual items are not used in the WebIQ project, they have no effect
- Example of usage of virtual items:
  - Lock: virtual:iMmProduction = virtual:EDH\_LOCKED
  - Hide: virtual:iMmProduction = virtual:EDH\_HIDDEN
- Adapt the group right text lists items, to the need of customers application, if requested

## 1.2 Setup of default group rights

In a customer project, the customer will define additional rights that do not exist in the template. The following steps are necessary to ensure that these are retained when updating to a new template version:

- Open WebIQ project where you defined additional group rights in WebIQ Designer
  - Open cfg.js in Code Manager
  - Save the changed arUserGroups[], arOpModes[] & arGrpData[] into a file (e.g. bak1.txt) see chapter 1.1 Implementation details
  - Login as admin
  - Go to page "Settings> Group rights"
  - Press button "Save config" => copies config string to Windows clipboard
  - Store the configuration from clipboard into file (e.g. bak2.txt)
- 
- Open new Template version in WebIQ Designer
  - Open cfg.js in Code Manager
  - Restore arUserGroups[], arOpModes[] & arGrpData[] from file (bak1.txt)
  - Copy group rights configuration from file bak2.txt to UI-Action groupRights.js.



```
actions["groupRights"] = async function (p) {  
  let sAction = p[0]  
  if (sAction === "save") { ...  
  } else if (sAction === "load-default") {  
    const jGrpRights = {  
      "operator": {  
        "iOpSetup": 3,  
        "iOpManual": 3,  
        "iSmItemWatchSetup": [0, 0, 0],  
        "iSmDebugConfig": [0, 0, 0]  
      },  
      "service": { ...  
    },  
    "admin": { ...  
  }  
}  
  let sRes = await cxt.msgBox("confirm", "Load default group rights?")
```

- Import all your application specific changes
- Save & publish project
- Run WebIQ project as admin
- Press button "Load default"
- Press button "Save config"