HEXAGON

HxGN EAM OpenCAD - White Paper
Ability to use RuleSet within EAM OC Grids

Document Release 12.21.0 2025, May File: Dashboard Rulesets Developer Guider03.docx
Written by: OpenCAD team
Approved by: Matteo Lodini

CONTENTS

Summary

INTRODUCTION

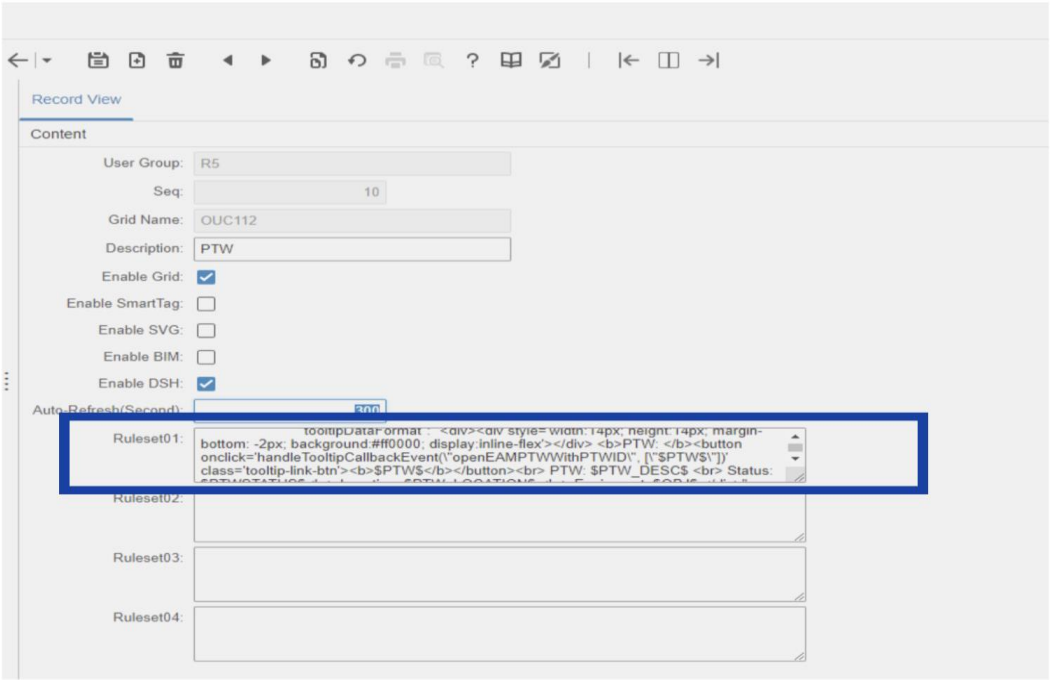
JGrids are a powerful yet configurable framework designed to give power in the hands of developers thanks to high customization, dataspyes, and user-defined rulesets. Rulesets are used to specify conditions based on a grid's columns and values.

WHAT CAN I DO WITH RULESETS

OC Rulesets allow to specify a set of rules in which any record can fall, and the graphical and informational layer which will be associated to any record falling inside a specific rule (See the use cases section for further details). In such a way admin users can pre-define for all users a set of relevant information to display on the dashboard, which will automatically be loaded on the document opening via OC Dashboard, without the need for additional user interaction, thus improving efficiency and adding value to an existing drawing.

HOW TO EDIT RULESETS

Rulesets configuration are grid-specific and can be reached by navigating through menu **OpenCAD Cross-Browser > Setup > Custom Grids**, and then selecting the desired grid. Rulesets appear in JSON format and can be edited as free text in UTF8 Encoding with max 2000 characters each. Up to 4 rulesets per grid can be defined.



RULESETS GRAMMAR

Rulesets have two main properties: “rulesetLabel”, which is used in the rulesets selection dropdown to inform users about the current setup which they are viewing, and “ruleset”, which will include an array of rules. The rational is the following: record after record is passed as in a sieve until it follows in any of the rulesets array.
Any ruleset in the “ruleset” key is recommended to have the following fields:

Key	Description	Format
(1)label	Is the label which appears as the name of the current group in the legend.	Free text
(2)condition	Is the condition which is evaluated to decide whether or not the current record should belong to the current ruleset group. The condition is evaluated by substituting the record’s corresponding property between COL_NAME in the condition statement	Free JS code
(3)tooltipDataFormat	Defines the format of the tooltip which will be rendered on the drawing when the user hovers an equipment. It is in free HTML text, to give maximum flexibility to developers to customize it.	Free HTML text
(4)symbol	Is the symbol which will appear on the drawing.	[“square”, “circle”, “rhombus”, “triangle-up”, “triangle-down”]. Default is “circle”
(5)color	Is the color of the symbol which will appear on the drawing.	Hex color format

USE CASE 0: DUMMY RULESET PTW GRID DEFAULT

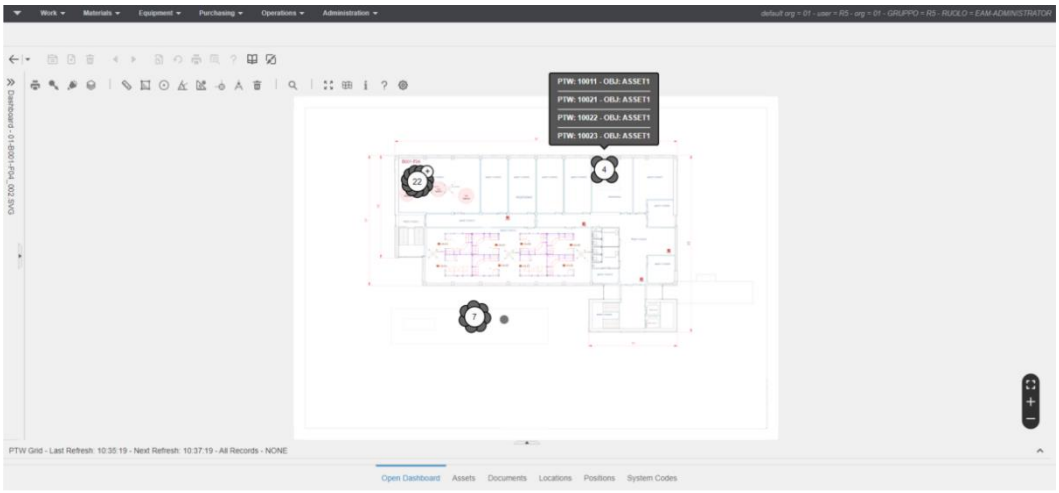
This dummy ruleset is the most simple one which involves Permit to works. It allows the user to see tags on the drawing which correspond to the object with Permit to works open on it.
With the mouse over on tag/flower will appear the smart tooltip.
Important note: rulesets are specific to a given custom grid since they refer to specific custom grid column names.

```
{
    "rulesetId": "1",
    "rulesetLabel": "NONE",
    "ruleset": [
        {
            "label": "Object",
            "tooltipDataFormat": "<div><b>PTW: $PTW$ - OBJ: $OBJ$</b></div>",
            "condition": "true",
            "symbol": "circle",
            "color": "#666666"
        }
    ]
}
```

The same JSON payload, once compressed, is the following:

```
{"rulesetId":"1","rulesetLabel":"NONE","ruleset":[{"label":"Object","tooltipDataFormat":"<div><b>PTW:  $PTW$
- OBJ: $OBJ$</b></div>","condition":"true","symbol":"circle","color":"#666666"}]}
```

Once inserted the rulset in the configuration OpenCAD Custom grid the result when we open the dashboard will be this:



USE CASE 1: SHOWING PTWS STATUS

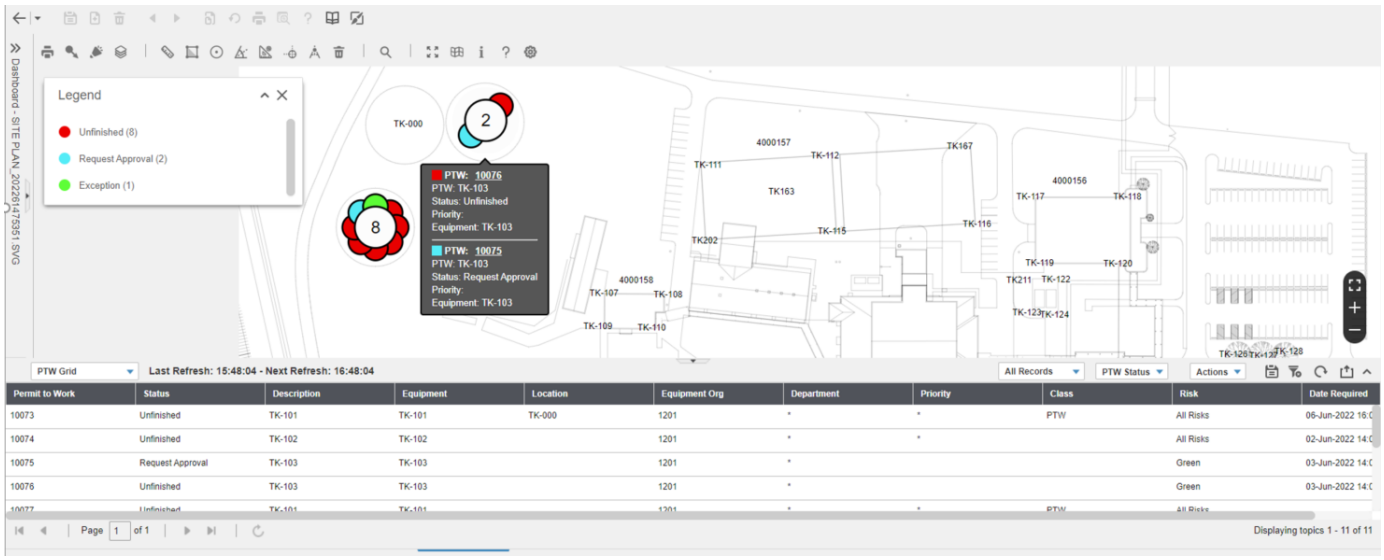
The aim of this use case is to display on the dashboard graphical information about the presence and localization of PTWs which have a STATUS Unfinished or Request Approval, depending on this, the record will appear on the drawing as different color, respectively, with a symbol and additional tooltip information.
If STATUS is different from 'Unfinished' or 'Request Approval' it is represented as 'Exception'.

N.B.: rulesets are specific to a given custom grid since they refer to specific custom grid column names. In the following case, a custom grid with the following columns is used:
["PTW_CODE","PTW_DESC", "OBJ", "OBJ_DESC", "ISP_TYPE", "ORG", "PTW_COMPLETED", "PTW_TYPEOFWORK", "PTW_STATUS", "PTW_TARGET", "PTW_LOTO", "OBJ_PARENT", "JOBTYPE"].

Below the rulesets JSON format

In this example three rules are visible: the last one, which has "condition": "true", is always verified, so any equipment which will not fall in any of the previous rules will then fall in this last rule (and appear as a green circle depending on the defined options).
It is good practice to compress the human readable JSON format before inserting it in the rulesets. This is a straightforward operation and can be easily done with many free internet tools, like (https://codebeautify.org/jsonminifier).
This will be the output result:


```
{"rulesetId":"1","rulesetLabel":"PTW
Status","ruleset":[{"label":"Unfinished","tooltipDataFormat":"<div><div style='width:14px; height:14px;
margin-bottom: -2px; background:#f70202; display:inline-flex'></div> <b>PTW: </b><button
onclick='handleTooltipCallbackEvent(\"openEAMPTWWithPTWID\", [\"$PTW$\"])' class='tooltip-link-
btn'><b>$PTW$</b></button><br> PTW: $PTW_DESC$ <br> Status: $PTWSTATUS$<br> Priority: $PTWPRIORITY$ <br>
Equipment: $OBJ$ </div>","condition":"$PTWSTATUS$ ==
'Unfinished'","symbol":"circle","color":"#f70202"},{"label":"Request
Approval","tooltipDataFormat":"<div><div style='width:14px; height:14px; margin-bottom: -2px;
background:#02ebf7; display:inline-flex'></div> <b>PTW: </b><button
onclick='handleTooltipCallbackEvent(\"openEAMPTWWithPTWID\", [\"$PTW$\"])' class='tooltip-link-
btn'><b>$PTW$</b></button><br> PTW: $PTW_DESC$ <br> Status: $PTWSTATUS$<br> Priority: $PTWPRIORITY$ <br>
Equipment: $OBJ$ </div>","condition":"$PTWSTATUS$ == 'Request
Approval'","symbol":"circle","color":"#02ebf7"},{"label":"Exception","tooltipDataFormat":"<div><div
style='width:14px; height:14px; margin-bottom: -2px; background:#00ff00; display:inline-flex'></div>
<b>PTW: </b><button onclick='handleTooltipCallbackEvent(\"openEAMPTWWithPTWID\", [\"$PTW$\"])'
class='tooltip-link-btn'><b>$PTW$</b></button><br> PTW: $PTW_DESC$ <br> Status: $PTWSTATUS$<br> Priority:
$PTWPRIORITY$ <br> Equipment: $OBJ$ </div>","condition":"true","symbol":"circle","color":"#00ff00"}]}
```


Dashboard - SITE PLAN_20261475551.SVG
Legend
Unfinished (8)
Request Approval (2)
Exception (1)
TK-000
2
8
PTW: 10076
PTW: TK-103
Status: Unfinished
Priority:
Equipment: TK-103
PTW: 10075
PTW: TK-103
Status: Request Approval
Priority:
Equipment: TK-103
4000157
TK-112
TK167
TK-111
TK163
4000156
TK-117
TK-118
TK-115
TK-116
TK202
TK-119
TK-120
TK211
TK-122
TK-123
TK-124
4000158
TK-107
TK-108
TK-109
TK-110
TK-128
PTW Grid
Last Refresh: 15:48:04 - Next Refresh: 16:48:04
All Records
PTW Status
Actions
Permit to Work
Status
Description
Equipment
Location
Equipment Org
Department
Priority
Class
Risk
Date Required
10073
Unfinished
TK-101
TK-101
TK-000
1201
PTW
All Risks
10074
Unfinished
TK-102
TK-102
1201
All Risks
10075
Request Approval
TK-103
TK-103
1201
Green
10076
Unfinished
TK-103
TK-103
1201
Green
Page 1 of 1
Displaying topics 1 - 11 of 11

USE CASE 2: SHOWING OEE AND TEXT OVERLAYS

The aim of this use case is to display OEE or Overlays Texts over a drawing, allowing users to display useful information coming from the EAM grids directly on the drawing without the need to open the tooltip. This is possible by specifying the parameter “hoverlayText" in a given rule. This field supports dynamic values that can be added by specifying the name of the corresponding column in the EAM table among “$” signs (i.e., “$OBJ_UDFCHAR11$”). In a given ruleset, it is possible to specify OEE and Overlays Texts also for a subset of rules and items.

```
{
  "rulesetId": "1",
  "rulesetLabel": "OEE",
  "ruleset": [
    {
      "label": "Equipment",
      "tooltipDataFormat": "<div><b>Code: $OBJ$ - Description: $OBJ_DESC$ - OEE:$OBJ_UDFCHAR11$ </b></div>",
      "condition": "parseInt($OBJ_UDFCHAR11$) >= 95",
      "symbol": "circle",
      "color": "#35a600",
      "hoverlayText": "$OBJ_UDFCHAR11$%"
    },
    {
      "label": "Equipment",
      "tooltipDataFormat": "<div><b>Code: $OBJ$ - Description: $OBJ_DESC$ - OEE:$OBJ_UDFCHAR11$ </b></div>",
      "condition": "parseInt($OBJ_UDFCHAR11$) >= 90",
      "symbol": "circle",
      "color": "#A020F0",
      "hoverlayText": "$OBJ_UDFCHAR11$%"
    },
    {
      "label": "Equipment",
      "tooltipDataFormat": "<div><b>Code: $OBJ$ - Description: $OBJ_DESC$ - OEE:$OBJ_UDFCHAR11$ </b></div>",
      "condition": "true",
      "symbol": "circle",
      "color": "#ff0000",
      "hoverlayText": "$OBJ_UDFCHAR11$%"
    }
  ]
}
```

N.B.: when the field “hoverlayText” is defined, the field “symbol” will be ignored.

Below the rulesets JSON format:


```
{"rulesetId":"1","rulesetLabel":"OEE","ruleset":[{"label":"Equipment","tooltipDataFormat":"<div><b>Code:
$OBJ$ - Description: $OBJ_DESC$ - OEE:$OBJ_UDFCHAR11$ </b></div>","condition":"parseInt($OBJ_UDFCHAR11$) >=
95”,"symbol":"circle", “color":"#35a600", “hoverlayText":"$OBJ_UDFCHAR11$%"},{"label":"Equipment",
"tooltipDataFormat":"<div><b>Code: $OBJ$ - Description: $OBJ_DESC$ - OEE:$OBJ_UDFCHAR11$ </b></div>”,
"condition":"parseInt($OBJ_UDFCHAR11$) >= 90”, “symbol":"circle", “color":"#A020F0",
“hoverlayText":"$OBJ_UDFCHAR11$%"},{"label":"Equipment", "tooltipDataFormat":"<div><b>Code: $OBJ$ -
Description: $OBJ_DESC$-OEE:$OBJ_UDFCHAR11$</b></div>”, “condition":"true", “symbol":"circle",
“color":"#ff0000", "hoverlayText":"$OBJ_UDFCHAR11$%"}]}
```