

```
1: /*
2: * File: Diagnostics.qml
3: * Description: Diagnostics screen that displays various engine information in a
4: *               scrollable table.
5: * Project: Carberry Pi
6: * Author: Ryan McHugh
7: * Year: 2020
8: */
9: import QtQuick 2.11
10: import QtQuick.Window 2.4
11: import QtQuick.Controls 2.4
12: import QtQuick.Controls.Styles 1.4
13: import QtQuick.Layouts 1.3
14: import QtQuick.Extras 1.4
15: import "../items" as Items
16:
17:
18: Frame{
19:     id: root_diagnostics
20:     property var context: null
21:     property alias props: diagnostics_model
22:     // property var time: context.time
23:
24:     property var ignore_list: {
25:         'connection-established' : true,
26:         'STATUS': true,
27:     }
28:
29:     clip: true
30:
31:     implicitWidth: 700
32:     implicitHeight: 420
33:
34:     // onTimeChanged: {console.log("hi")}
35:
36:     ListView {
37:
38:         anchors.horizontalCenter: parent.horizontalCenter
39:         anchors.verticalCenter: parent.verticalCenter
40:
41:         id: diagnostics_view
42:
43:         implicitHeight: root_diagnostics.implicitHeight - 50
44:         implicitWidth: root_diagnostics.implicitWidth - 50
45:
46:         clip: true
47:
48:         model: ListModel {
49:             id: diagnostics_model
50:             dynamicRoles: true
51:
52:
53:             function refreshModel(){
54:
55:                 clear()
56:
57:                 // console.log(ignore_list)
58:
59:                 var obj = Object.keys(context.diagnostics);
60:
61:                 for(var i = 0; i < obj.length; i++){
62:                     var fill = obj[i];
63:                     var entry = {'key': fill, 'value': context.diagnostics[fill]};
64:                     if(ignore_list[fill] == undefined && context.diagnostics[fill] !=
[ ])
65:                         this.append(entry);
66:                 }
67:
```

```
68:
69:         if(this.count == 0)
70:             this.append({'key': "Response", 'value': "No Content to Display"})
71:
72:
73:     }
74:
75:     Component.onCompleted: refreshModel()
76:
77: }
78:
79:
80:
81:
82:     delegate: diagnosticsDelegate
83:
84:     Component {
85:         id: diagnosticsDelegate
86:
87:         Item{
88:             id: diagnosticsItem
89:             width: diagnostics_view.implicitWidth
90:
91:             function get_preferred_height(){
92:                 var w_key = model.key.length
93:                 // var w_val = context.diagnostics[model.key].toString().length
94:                 var w_val = model.value.length
95:                 var ret = '';
96:                 if(w_key > w_val)
97:                     ret = w_key;
98:                 else
99:                     ret = w_val;
100:                 // return (Math.pow(40, ret / (diagnosticsItem.width / 2) + 1)) + 20
;
101:                 ret = (ret / 40) * 45;
102:                 return ret > 50 ? ret : 50;
103:             }
104:
105:             height: get_preferred_height();
106:             clip: true
107:
108:             Items.Rectangle{
109:                 style: context.config['style']['current']
110:                 width: parent.width
111:                 height: parent.height - 10
112:
113:                 Row{
114:                     anchors.fill: parent
115:
116:                     Column{
117:                         anchors.rightMargin: 25
118:                         clip: true
119:                         height: parent.height
120:                         width: parent.width / 2
121:                         Items.Label {
122:                             text: model.key
123:                             width: parent.width
124:                             style: context.config['style']['current']
125:                         }
126:                     }
127:
128:                     Items.Rectangle{
129:                         anchors.leftMargin: 10
130:                         anchors.rightMargin: 10
131:                         anchors.verticalCenter: parent.verticalCenter
132:                         width: 5;
133:                         height: parent.height - 10;
134:                         style: "inverted_" + context.config['style']['current']
```

```
135:         }
136:
137:         Column{
138:             anchors.leftMargin: 25
139:             clip: true
140:             height: parent.height
141:             width: parent.width / 2
142:             Items.Label {
143:                 text: model.value
144:                 width: parent.width
145:                 style: context.config['style']['current']
146:             }
147:         }
148:     }
149: }
150:
151: }
152:
153: }
154:
155: }
156:
157: // Row{
158: //     anchors.top: diagnostics_view.bottom
159: //     anchors.topMargin: 20
160: //     anchors.horizontalCenter: parent.horizontalCenter
161: //     Column{
162: //         Items.Button{
163: //             text: "Refresh"
164: //             style: context.config['style']['current']
165: //             onClicked: function(){
166: //                 // setTimeout(function(){""}, 1000)
167: //                 diagnostics_model.refreshModel()
168: //             }
169: //         }
170: //     }
171: // }
172:
173:
174: }
175:
```