MeshAgent / modules / win-info.js

260 lines (236 loc) · 9.16 KB

Code Blame Raw

```
/*
Copyright 2019-2020 Intel Corporation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/


//
// This module fetches various Windows System information, such as pending reboot status,
// volume defrag state, installed applications, windows update status, etc
//


var promise = require('promise');

//
```

```
27      // This function queries WMI to fetch Windows Update Status
28      //
29      function qfe()
30      {
31          var child = require('child_process').execFile(process.env['windir'] + '\\System32\\wbem\\wmic.e
32          child.stdout.str = ''; child.stdout.on('data', function (c) { this.str += c.toString(); });
33          child.stderr.str = ''; child.stderr.on('data', function (c) { this.str += c.toString(); });
34          child.waitExit();
35
36          var lines = child.stdout.str.trim().split('\r\n');
37          var keys = lines[0].split(',');
38          var i, key;
39          var tokens;
40          var result = [];
41
42          for (i = 1; i < lines.length; ++i)
43          {
44              var obj = {};
45              tokens = lines[i].split(',');
46              for (key = 0; key < keys.length; ++key)
47              {
48                  if (tokens[key]) { obj[keys[key]] = tokens[key]; }
49              }
50              result.push(obj);
51          }
52          return (result);
53      }
54
55      // This function uses Windows Powershell to fetch metadata about the currently configured AntiVirus
56      function av()
57      {
58          var child = require('child_process').execFile(process.env['windir'] + '\\System32\\WindowsPower
59          if (child == null) { return ([]); }
60
61          child.descriptorMetadata = 'process-manager';
62          child.stdout.str = ''; child.stdout.on('data', function (c) { this.str += c.toString(); });
63          child.stderr.str = ''; child.stderr.on('data', function (c) { this.str += c.toString(); });
64
65          child.stdin.write('[reflection.Assembly]::LoadWithPartialName("system.core")\r\n');
66          child.stdin.write('Get-WmiObject -Namespace "root/SecurityCenter2" -Class AntiVirusProduct | ')
67          child.stdin.write('ForEach-Object -Process { ');
68          child.stdin.write('$Bytes = [System.Text.Encoding]::UTF8.GetBytes($_.displayName); ');
69          child.stdin.write('$EncodedText =[Convert]::ToBase64String($Bytes); ');
70          child.stdin.write('Write-Host ("{0},{1}" -f $_.productState,$EncodedText); }\r\n');
71          child.stdin.write('exit\r\n');
72          child.waitExit();
```

```
73
74          if (child.stdout.str == '') { return ([]); }
75
76          var lines = child.stdout.str.trim().split('\r\n');
77          var result = [];
78          for (i = 0; i < lines.length; ++i)
79          {
80              var keys = lines[i].split(',');
81              if(keys.length == 2)
82              {
83                  var status = {};
84                  status.product = Buffer.from(keys[1], 'base64').toString();
85                  status.updated = (parseInt(keys[0]) & 0x10) == 0;
86                  status.enabled = (parseInt(keys[0]) & 0x1000) == 0x1000;
87                  result.push(status);
88              }
89          }
90          return (result);
91      }
92
93      //
94      // This function uses the defrag system utility to query defrag state of the specified volume
95      //
96      // Note: options.volume must be specified
97      function defrag(options)
98      {
99          var ret = new promise(function (res, rej) { this._res = res; this._rej = rej; });
100         var path = '';
101
102         switch(require('os').arch())
103         {
104             case 'x64':
105                 if (require('_GenericMarshal').PointerSize == 4)
106                 {
107                     // 32 Bit App on 64 Bit Windows
108                     ret._rej('Cannot defrag volume on 64 bit Windows from 32 bit application');
109                     return (ret);
110                 }
111                 else
112                 {
113                     // 64 Bit App
114                     path = process.env['windir'] + '\\System32\\defrag.exe';
115                 }
116                 break;
117             case 'ia32':
118                 // 32 Bit App on 32 Bit Windows
```

```
187         {
188             ret = 'Windows Update';
189         }
190         else if ((tmp=regQuery(HKEY.LocalMachine, 'SYSTEM\\CurrentControlSet\\Control\\Session Manager'
191         {
192             ret = 'File Rename';
193         }
194         else if (regQuery(HKEY.LocalMachine, 'SYSTEM\\CurrentControlSet\\Control\\ComputerName\\ActiveC
195         {
196             ret = 'System Rename';
197         }
198         return (ret);
199     }
200 
201     //
202     // Returns a promise that fetches the list of installed applications
203     //
204     function installedApps()
205     {
206         var promise = require('promise');
207         var ret = new promise(function (a, r) { this._resolve = a; this._reject = r; });
208 
209         var code = "\
```

```
210         var reg = require('win-registry');\
211         var result = [];\
212         var val, tmp;\
213         var items = reg.QueryKey(reg.HKEY.LocalMachine, 'SOFTWARE\\\\Microsoft\\\\Windows\\\\CurrentVer
214         for (var key in items.subkeys)\
215         {\
216             val = {};\
217             try\
218             {\
219                 val.name = reg.QueryKey(reg.HKEY.LocalMachine, 'SOFTWARE\\\\Microsoft\\\\Windows\\\\Cur
220             }\
221             catch(e)\
222             {\
223                 continue;\
224             }\
225             try\
226             {\
227                 val.version = reg.QueryKey(reg.HKEY.LocalMachine, 'SOFTWARE\\\\Microsoft\\\\Windows\\\\
228                 if (val.version == '') { delete val.version; }\
229             }\
230             catch(e)\
231             {\
232             }\
233             try\
234             {\
235                 val.location = reg.QueryKey(reg.HKEY.LocalMachine, 'SOFTWARE\\\\Microsoft\\\\Windows\\\
236                 if (val.location == '') { delete val.location; }\
237             }\
238             catch(e)\
239             {\
240             }\
241             result.push(val);\
242         }\
243         console.log(JSON.stringify(result,'', 1));process.exit();";
244
245         ret.child = require('child_process').execFile(process.execPath, [process.execPath.split('\\').p
246         ret.child.promise = ret;
247         ret.child.stdout.str = ''; ret.child.stdout.on('data', function (c) { this.str += c.toString();
248         ret.child.on('exit', function (c) { this.promise._resolve(JSON.parse(this.stdout.str.trim()));
249         return (ret);
250     }
251
252     if (process.platform == 'win32')
253     {
254         module.exports = { qfe: qfe, av: av, defrag: defrag, pendingReboot: pendingReboot, installedApp
255     }
```

```
256     else
257     {
258         var not_supported = function () { throw (process.platform + ' not supported'); };
259         module.exports = { qfe: not_supported, av: not_supported, defrag: not_supported, pendingReboot:
260     }
```