1-iloRestBasics

January 27, 2021

1 iLOrest: The HPE Redfish Swiss knife

Version 0.50

1.1 What is iLOrest?

- Command line Redfish client primarily designed to managed HPE iLO 4 and iLO 5 based servers
- Python oriented
- Sources on GitHub (python-redfish-utility)
- Packaged binaries available for several Operating Environments (.deb, .rpm, .msi and .pkg)
- Windows and RPM packages available on HPE Customer Support Center

1.1.1 Extreme flexibility

- Interactive mode allows you to save and load settings from a file with command completion (tab key) and command recall (up and down arrows)
- Script mode: easy integration with bash, PowerShell or DOS
- File-based mode: allows you to save, customize and then deploy settings
- Extensible (See tutorial video)
- In-band and out-of-band management
- Useful for debugging Python and PowerShell scripts
- Rich ecosystem with videos and articles

1.2 Interactive out-of-band mode demo

1.2.1 Command completion and recall with help description

Open a `Terminal` from a Jupyter Launcher and issue the following commands. To ease this exercise, you can open a New View for this notebook by right clicking on its tabulation:

```
# Invoke iLOrest with no arguments
ilorest

# Type beginning of command and hit the Tab key to complete the command
ilorest > he (tab)

# Recall help command with up-arrow
# login remote iLO as user student. If password is not suppplied, iLOrest asks for password in h
iLOrest > login ilo5 -u student
password: P@ssw0rd!
```

1.2.2 Data/Resource Types

Redfish resources are associated with a data type. The list of all iLO 5 data types can be retrieved with the `types` command. Each data type and its content is present in the API Reference document.

```
# List all data types
iLOrest > types
```

Data types with prefix `Hpe` are Oem/Hpe specific types. Others are standard Redfish types. The following cell lists all data types present in the Redfish service, selects the `Bios` data type and shows all the resources associated with that type.

```
# Select the Bios type
iLOrest > select Bio(Tab)

# View selected type
iLOrest > select

# Show resources associated to the selected type
iLOrest > get

# Retrieve a single reseource from another type in json format
iLOrest > get SecurityState --json --select HpeSecurityService.

# Logout: close session. Local cache will be removed automatically.
iLOrest > logout
iLOrest > exit
```

1.2.3 In-band management demo: Modification of a BIOS attribute

> **NOTE**: Only a **privileged instructor** can perform the following commands

```
ssh root@ilo5server

# When logged as root and when iLO 5 security mode/state is Production, no need to provide crede
ilorest login

# Retrieve and modify Bios parameter
ilorest get AdminName --select Bios.
ilorest set AdminName="John Deuf"
ilorest status --json    # Useful for rawpatch command

# Commit modification
ilorest commit
```

```
# Reset cache and view modification
ilorest select Bios. --refresh
ilorest get AdminName

# Logout (complete cache removal)
ilorest logout
```

1.3 Script mode examples

1.3.1 Environment preparation

The following cell sets environment variables and checks the connectivity toward the various BMCs used in this notebook.

[1]:

```
########## Environment preparation (Version: 0.26) ############


# Set Student ID number
Stud=675
Id=$(id --user --name)

# location and ports variables
ObmcBasePort=44000
IloDlBasePort=45000
IloSyBasePort=46000
let OpenBmcBasePort=$ObmcBasePort
let OpenBmcPort=${OpenBmcBasePort}+${Stud}
let iLO5SimulatorBasePort=$IloDlBasePort
let iLO5SimulatorPort=${iLO5SimulatorBasePort}+${Stud}
let ilo5Port=443

CacheDir="${PWD}/iLOrestCache"
LogDir="${PWD}"
LogFile="${LogDir}/iLOrest.log"

#iLO5SimulatorIP=bmcsimulators
iLO5SimulatorIP=ilo5simulators
iLOSimulator=${iLO5SimulatorIP}:${iLO5SimulatorPort}
iLO5SimulatorURI=https://${iLOSimulator}

OpenBmcIP=openbmcsimulators
OpenBmc="${OpenBmcIP}:${OpenBmcPort}"
OpenBmcURI="https://${OpenBmc}"

ilo5IP="ilo5"
ilo5="${ilo5IP}:${ilo5Port}"
```

```
ilo5URI="https://${ilo5}"

# Credentials
User="student"
Password='P@ssw0rd!'

# Miscellaneous
alias ilorest='ilorest --nologo --cache-dir ${CacheDir} --logdir ${LogDir}'
alias ResetIlo5Simulator="ssh rstsimul@${iLO5SimulatorIP} $Id␣
 ↪$iLO5SimulatorBasePort"
ilorest logout &>/dev/null # Clear cache

# Verify we can reach the remote Bmcs on the right HTTPS ports.
for bmc in ilo5 OpenBmc iLO5Simulator ; do
    ip="${bmc}IP" ; port=$(echo ${bmc}Port)
    nc -vz  $(eval echo "\$${ip}") $(eval echo "\$${port}") &> /dev/null &&
        echo "$bmc is reachable" \
        || echo "WARNING: Problem reaching $bmc"
done
```

```
ilo5 is reachable
OpenBmc is reachable
iLO5Simulator is reachable
```

1.3.2 Log into a real iLO 5 to populate the cache directory

iLOrest populates a cache directory during the execution of the `login` command just after a successul authentication. However, the DMTF Redfish simulator used in this workshop does not implement authentication and returns an error to the `login` command, preventing iLOrest to populate its cache.

To overcome this problem you will first log into a physical iLO 5 and then hack the cache to make it point to the DMTF iLO 5 simulator.

[2]:

```
ilorest login $ilo5 -u $User -p $Password
head ${CacheDir}/cache/* | grep url | sort -u                    # extract URL␣
 ↪target in cache

# Copy cache in a safe location before logging out to
# minimize opened sessions. iLO 5 is limited to 15 concurrent
# sessions.
mv ${CacheDir}/cache{,-bck}
ilorest logout

# Move back the cache
mv ${CacheDir}/cache{-bck,}

# Redirect cache to point to the iLO 5 Simulator
```

```
sed -i "s.${ilo5URI}.${iLO5SimulatorURI}." ${CacheDir}/cache/*
head ${CacheDir}/cache/* | grep url | sort -u                    # extract URL target in cache
```

```
Discovering data...Done
    "url": "https://ilo5:443",
Logging session out.
    "url": "https://ilo5simulators:45675",
```

1.4 Get multiple properties with single request

The following request retrieves all firmware versions and description with a single command. Note that DMTF includes computer firmware in the `SoftwareInventory` collection.

> **NOTE**: iLOrest versions 3.0, 3.1.0 and 3.1.1 return an `ERROR: list index out of range` message.

[3]:
```
ilorest --version

ilorest get Name Version --select SoftwareInventory
```

```
RESTful Interface Tool 3.1.1
ERROR: list index out of range
```

1.4.1 Change iLO time zone

Searching for keywords `time zone` or `timezone` in the API Reference document leads to **two** properties. A BIOS and an iLO related property. We will focus on the iLO property.

The iLO `TimeZone` property is associated to the `HpeiLODateTime` type. The next cells select this data type and then modify it.

[4]:
```
# select type
ilorest select HpeiLODateTime

# Get all resources from the selected data type in stripped mode
ilorest get --json
```

```
{
  "ConfigurationSettings": "Current",
  "DateTime": "2020-12-11T11:04:22Z",
  "NTPServers": [
    "",
    ""
  ],
  "PropagateTimeToHost": false,
"StaticNTPServers": [
  "0.0.0.0",
  "0.0.0.0"
],
"TimeZone": {
  "Index": 15,
  "UtcOffset": "+00:00",
  "Value": "GMT-0"
},
"TimeZoneList": [
  {
    "Index": 0,
    "UtcOffset": "-12:00",
    "Value": "GMT+12:00"
  },
  {
    "Index": 1,
    "UtcOffset": "-11:00",
    "Value": "SST+11:00"
  },
  {
    "Index": 2,
    "UtcOffset": "-10:00",
    "Value": "HST+10:00"
  },
  {
    "Index": 3,
    "UtcOffset": "-09:30",
    "Value": "MART+9:30"
  },
  {
    "Index": 4,
    "UtcOffset": "-09:00",
    "Value": "AKST+9:00AKDT+08:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
  },
  {
    "Index": 5,
    "UtcOffset": "-08:00",
    "Value": "PST+8:00PDT+07:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
  },
  {
    "Index": 6,
    "UtcOffset": "-07:00",
    "Value": "MST+7:00MDT+06:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
  },
  {
    "Index": 7,
    "UtcOffset": "-06:00",
      "Value": "CST+6:00CDT+05:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
    },
    {
      "Index": 8,
      "UtcOffset": "-05:00",
      "Value": "EST+5:00EDT+04:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
    },
    {
      "Index": 9,
      "UtcOffset": "-04:00",
      "Value": "VET+4:00"
    },
    {
      "Index": 10,
      "UtcOffset": "-04:00",
      "Value": "AST+4:00ADT+03:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
    },
    {
      "Index": 11,
      "UtcOffset": "-03:30",
      "Value": "NST+3:30NDT+02:30:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
    },
    {
      "Index": 12,
      "UtcOffset": "-03:00",
      "Value": "ART+3:00"
    },
    {
      "Index": 13,
      "UtcOffset": "-02:00",
      "Value": "GST+2:00"
    },
    {
      "Index": 14,
      "UtcOffset": "-01:00",
      "Value": "CVT+1:00"
    },
    {
      "Index": 15,
      "UtcOffset": "+00:00",
      "Value": "GMT-0"
    },
    {
      "Index": 16,
      "UtcOffset": "+01:00",
      "Value": "CET-1:00CEST-02:00:00,M3.5.0/01:00:00,M10.5.0/01:00:00"
    },
    {
```

```
  "Index": 17,
  "UtcOffset": "+02:00",
  "Value": "EET-2:00EEST-03:00:00,M3.5.0/01:00:00,M10.5.0/01:00:00"
},
{
  "Index": 18,
  "UtcOffset": "+03:00",
  "Value": "AST-3:00"
},
{
  "Index": 19,
  "UtcOffset": "+03:30",
  "Value": "IRST-3:30IRDT-04:30:00,80/00:00:00,264/00:00:00"
},
{
  "Index": 20,
  "UtcOffset": "+04:00",
  "Value": "GST-4:00"
},
{
  "Index": 21,
  "UtcOffset": "+04:30",
  "Value": "AFT-4:30"
},
{
  "Index": 22,
  "UtcOffset": "+05:00",
  "Value": "YEKT-5:00"
},
{
  "Index": 23,
  "UtcOffset": "+05:30",
  "Value": "IST-5:30"
},
{
  "Index": 24,
  "UtcOffset": "+05:45",
  "Value": "NPT-5:45"
},
{
  "Index": 25,
  "UtcOffset": "+06:00",
  "Value": "ALMT-6:00"
},
{
  "Index": 26,
  "UtcOffset": "+06:30",
  "Value": "MMT-6:30"
```

```
},
{
  "Index": 27,
  "UtcOffset": "+07:00",
  "Value": "ICT-7:00"
},
{
  "Index": 28,
  "UtcOffset": "+08:00",
  "Value": "CST-8:00"
},
{
  "Index": 29,
  "UtcOffset": "+08:45",
  "Value": "ACWST-08:45"
},
{
  "Index": 30,
  "UtcOffset": "+09:00",
  "Value": "JST-9:00"
},
{
  "Index": 31,
  "UtcOffset": "+09:30",
  "Value": "ACST-9:30ACDT-10:30:00,M10.1.0/02:00:00,M4.1.0/02:00:00"
},
{
  "Index": 32,
  "UtcOffset": "+10:00",
  "Value": "AEST-10:00AEDT-11:00:00,M10.1.0/02:00:00,M4.1.0/02:00:00"
},
{
  "Index": 33,
  "UtcOffset": "+10:30",
  "Value": "LHST-10:30LHDT11:00"
},
{
  "Index": 34,
  "UtcOffset": "+10:45",
  "Value": "CHAST-10:45CHADT-11:45"
},
{
  "Index": 35,
  "UtcOffset": "+11:00",
  "Value": "MAGT-11:00"
},
{
  "Index": 36,
      "UtcOffset": "+12:00",
      "Value": "NZST-12:00NZDT-13:00:00,M9.5.0/02:00:00,M4.1.0/02:00:00"
    },
    {
      "Index": 37,
      "UtcOffset": "+13:00",
      "Value": "TKT-13:00"
    },
    {
      "Index": 38,
      "UtcOffset": "+14:00",
      "Value": "LINT-14:00"
    },
    {
      "Index": 39,
      "UtcOffset": "+00:00",
      "Value": "GMT-0"
    }
  ]
}
```

[5]:
```
# Get all resources associated to this data type in verbose mode
ilorest list --json
```

```
{
  "@odata.context": "/redfish/v1/$metadata#HpeiLODateTime.HpeiLODateTime",
  "@odata.etag": "W/\"09805A7E\"",
  "@odata.id": "/redfish/v1/Managers/1/DateTime/",
  "@odata.type": "#HpeiLODateTime.v2_0_0.HpeiLODateTime",
  "ConfigurationSettings": "Current",
  "DateTime": "2020-12-11T11:04:22Z",
  "Id": "DateTime",
  "Links": {
    "EthernetNICs": {
      "@odata.id": "/redfish/v1/Managers/1/EthernetInterfaces/"
    }
  },
  "NTPServers": [
    "",
    ""
  ],
  "Name": "iLO Date and Time Settings",
  "PropagateTimeToHost": false,
  "StaticNTPServers": [
    "0.0.0.0",
    "0.0.0.0"
  ],
  "TimeZone": {
    "Index": 15,
    "Name": "Greenwich Mean Time, Casablanca, Monrovia, Dublin, London",
    "UtcOffset": "+00:00",
    "Value": "GMT-0"
  },
  "TimeZoneList": [
    {
      "Index": 0,
      "Name": "International Date Line West",
      "UtcOffset": "-12:00",
      "Value": "GMT+12:00"
    },
    {
      "Index": 1,
      "Name": "Midway Island, Samoa",
      "UtcOffset": "-11:00",
      "Value": "SST+11:00"
    },
    {
      "Index": 2,
      "Name": "Hawaii",
      "UtcOffset": "-10:00",
      "Value": "HST+10:00"
    },
    {
      "Index": 3,
      "Name": "Marquesas",
      "UtcOffset": "-09:30",
      "Value": "MART+9:30"
    },
    {
      "Index": 4,
      "Name": "Alaska",
      "UtcOffset": "-09:00",
      "Value": "AKST+9:00AKDT+08:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
    },
    {
      "Index": 5,
      "Name": "Pacific Time(US & Canada), Tijuana, Portland",
      "UtcOffset": "-08:00",
      "Value": "PST+8:00PDT+07:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
    },
    {
      "Index": 6,
      "Name": "Arizona, Chihuahua, La Paz, Mazatlan, Mountain Time (US & Canad",
      "UtcOffset": "-07:00",
      "Value": "MST+7:00MDT+06:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
    },
{
  "Index": 7,
  "Name": "Central America, Central Time(US & Canada)",
  "UtcOffset": "-06:00",
  "Value": "CST+6:00CDT+05:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
},
{
  "Index": 8,
  "Name": "Bogota, Lima, Quito, Eastern Time(US & Canada)",
  "UtcOffset": "-05:00",
  "Value": "EST+5:00EDT+04:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
},
{
  "Index": 9,
  "Name": "Caracas, Georgetown",
  "UtcOffset": "-04:00",
  "Value": "VET+4:00"
},
{
  "Index": 10,
  "Name": "Atlantic Time(Canada), Santiago",
  "UtcOffset": "-04:00",
  "Value": "AST+4:00ADT+03:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
},
{
  "Index": 11,
  "Name": "Newfoundland",
  "UtcOffset": "-03:30",
  "Value": "NST+3:30NDT+02:30:00,M3.2.0/02:00:00,M11.1.0/02:00:00"
},
{
  "Index": 12,
  "Name": "Brasilia, Buenos Aires, Greenland",
  "UtcOffset": "-03:00",
  "Value": "ART+3:00"
},
{
  "Index": 13,
  "Name": "Mid-Atlantic",
  "UtcOffset": "-02:00",
  "Value": "GST+2:00"
},
{
  "Index": 14,
  "Name": "Azores, Cape Verde Is.",
  "UtcOffset": "-01:00",
  "Value": "CVT+1:00"
},
{
  "Index": 15,
  "Name": "Greenwich Mean Time, Casablanca, Monrovia, Dublin, London",
  "UtcOffset": "+00:00",
  "Value": "GMT-0"
},
{
  "Index": 16,
  "Name": "Amsterdam, Berlin, Bern, Rome, Paris, West Central Africa",
  "UtcOffset": "+01:00",
  "Value": "CET-1:00CEST-02:00:00,M3.5.0/01:00:00,M10.5.0/01:00:00"
},
{
  "Index": 17,
  "Name": "Athens, Bucharest, Cairo, Jerusalem",
  "UtcOffset": "+02:00",
  "Value": "EET-2:00EEST-03:00:00,M3.5.0/01:00:00,M10.5.0/01:00:00"
},
{
  "Index": 18,
  "Name": "Baghdad, Kuwait, Riyadh, Moscow, Istanbul, Nairobi",
  "UtcOffset": "+03:00",
  "Value": "AST-3:00"
},
{
  "Index": 19,
  "Name": "Tehran",
  "UtcOffset": "+03:30",
  "Value": "IRST-3:30IRDT-04:30:00,80/00:00:00,264/00:00:00"
},
{
  "Index": 20,
  "Name": "Abu Dhabi, Muscat, Baku, Tbilisi, Yerevan",
  "UtcOffset": "+04:00",
  "Value": "GST-4:00"
},
{
  "Index": 21,
  "Name": "Kabul",
  "UtcOffset": "+04:30",
  "Value": "AFT-4:30"
},
{
  "Index": 22,
  "Name": "Ekaterinburg, Islamabad, Karachi, Tashkent",
  "UtcOffset": "+05:00",
  "Value": "YEKT-5:00"
},
{
  "Index": 23,
  "Name": "Chennai, Kolkata, Mumbai, New Delhi",
  "UtcOffset": "+05:30",
  "Value": "IST-5:30"
},
{
  "Index": 24,
  "Name": "Kathmandu",
  "UtcOffset": "+05:45",
  "Value": "NPT-5:45"
},
{
  "Index": 25,
  "Name": "Almaty, Dhaka, Sri Jayawardenepura",
  "UtcOffset": "+06:00",
  "Value": "ALMT-6:00"
},
{
  "Index": 26,
  "Name": "Rangoon",
  "UtcOffset": "+06:30",
  "Value": "MMT-6:30"
},
{
  "Index": 27,
  "Name": "Bangkok, Hanio, Jakarta, Novosibirsk, Astana, Krasnoyarsk",
  "UtcOffset": "+07:00",
  "Value": "ICT-7:00"
},
{
  "Index": 28,
  "Name": "Beijing, Chongqing, Hong Kong, Urumqi, Taipei, Perth",
  "UtcOffset": "+08:00",
  "Value": "CST-8:00"
},
{
  "Index": 29,
  "Name": "Eucla",
  "UtcOffset": "+08:45",
  "Value": "ACWST-08:45"
},
{
  "Index": 30,
  "Name": "Osaka, Sapporo, Tokyo, Seoul, Yakutsk",
  "UtcOffset": "+09:00",
  "Value": "JST-9:00"
},
{
  "Index": 31,
  "Name": "Adelaide, Darwin",
  "UtcOffset": "+09:30",
  "Value": "ACST-9:30ACDT-10:30:00,M10.1.0/02:00:00,M4.1.0/02:00:00"
},
{
  "Index": 32,
  "Name": "Canberra, Melbourne, Sydney, Guam, Hobart, Vladivostok",
  "UtcOffset": "+10:00",
  "Value": "AEST-10:00AEDT-11:00:00,M10.1.0/02:00:00,M4.1.0/02:00:00"
},
{
  "Index": 33,
  "Name": "Lord Howe",
  "UtcOffset": "+10:30",
  "Value": "LHST-10:30LHDT11:00"
},
{
  "Index": 34,
  "Name": "Chatham",
  "UtcOffset": "+10:45",
  "Value": "CHAST-10:45CHADT-11:45"
},
{
  "Index": 35,
  "Name": "Magadan, Solomon Is., New Caledonia",
  "UtcOffset": "+11:00",
  "Value": "MAGT-11:00"
},
{
  "Index": 36,
  "Name": "Auckland, Wellington, Fiji, Kamchatka, Marshall Is.",
  "UtcOffset": "+12:00",
  "Value": "NZST-12:00NZDT-13:00:00,M9.5.0/02:00:00,M4.1.0/02:00:00"
},
{
  "Index": 37,
  "Name": "Nuku'alofa",
  "UtcOffset": "+13:00",
  "Value": "TKT-13:00"
},
{
  "Index": 38,
  "Name": "Line Islands",
  "UtcOffset": "+14:00",
  "Value": "LINT-14:00"
},
    {
      "Index": 39,
      "Name": "Unspecified Time Zone",
      "UtcOffset": "+00:00",
      "Value": "GMT-0"
    }
  ]
}
```

[6]:

```
# You can get more info about a specific property.
# Note the "READ-ONLY" attribute.
ilorest info TimeZone
```

```
NAME
    TimeZone


DESCRIPTION
    The currently selected time zone.


TYPE
    object


READ-ONLY
    False


SUB-PROPERTIES
    Index, UtcOffset, Name, Value
```

[7]:

```
# Set iLO TimeZone to "New Delhi"
ilorest set TimeZone/Name="New Delhi"

# View changed settings
ilorest status
```

```
Current changes found:
HpeiLODateTime.v2_0_0(/redfish/v1/Managers/1/DateTime/) (Currently selected)
        TimeZone/Name=New Delhi
```

[8]:

```
# Commit changes in Redfish server
ilorest commit
```

```
Committing changes...
The operation completed successfully.
```

> **NOTE**: A successful commit against a real iLO 5 clears pending changes in the cache. However, our simulator does not behave like a real iLO 5 and we need to clean up the cache manually to mimic a real successful commit.

[9]:

```
ilorest status
ilorest select HpeiLODateTime --refresh
ilorest status
```

```
Current changes found:
HpeiLODateTime.v2_0_0(/redfish/v1/Managers/1/DateTime/) (Currently selected)
        TimeZone/Name=New Delhi
No changes found
```

1.4.2 Use of the –filter option

In this example you will use the iLOrest `--filter` option to retrieve only the IPv4 address of the iLO Dedicated Network Port.

Searching for keywords `dedicated` or `shared` in the API Reference document leads to the `EthernetInterface` data type. The following cell logs into a remote iLO 5 and retrieves all the data types containing the string `Ethernet`.

[10]:

```
ilorest types | grep Ethernet
```

```
EthernetInterface.v1_4_1
EthernetInterfaceCollection
```

Data types with suffix `Collection` group similar resources usually represented in an array of `Members` links. The following cell lists all the ethernet interfaces in a system: System Interfaces (4) and Manager Interfaces (3).

[11]:

```
ilorest list --json --select EthernetInterfaceCollection
```

```
[
  {
    "@odata.context":
"/redfish/v1/$metadata#EthernetInterfaceCollection.EthernetInterfaceCollection",
    "@odata.etag": "W/\"96E00A2C\"",
    "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/",
    "@odata.type": "#EthernetInterfaceCollection.EthernetInterfaceCollection",
    "Description": "Collection of System Ethernet Interfaces",
    "Members": [
      {
        "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/1/"
      },
      {
        "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/2/"
      },
      {
        "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/3/"
      },
      {
        "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/4/"
      }
    ],
    "Members@odata.count": 4,
    "Name": "System Ethernet Interfaces"
  },
  {
    "@odata.context":
"/redfish/v1/$metadata#EthernetInterfaceCollection.EthernetInterfaceCollection",
    "@odata.etag": "W/\"E589C4BF\"",
    "@odata.id": "/redfish/v1/Managers/1/EthernetInterfaces/",
    "@odata.type": "#EthernetInterfaceCollection.EthernetInterfaceCollection",
    "Description": "Configuration of Manager Network Interfaces",
    "Members": [
      {
        "@odata.id": "/redfish/v1/Managers/1/EthernetInterfaces/1/"
      },
      {
        "@odata.id": "/redfish/v1/Managers/1/EthernetInterfaces/2/"
      },
      {
        "@odata.id": "/redfish/v1/Managers/1/EthernetInterfaces/3/"
      }
    ],
    "Members@odata.count": 3,
    "Name": "Manager Network Interfaces"
  }
]
```

To retrieve the IPv4 address of the dedicated iLO network port you need to find a unique property characterizing the iLO Dedicated Network Port. The following cells select the EthernetInterface data type and prints the Name property of each and every ethernet port in the server to verify whether it is a good candidate for a filter criteria.

[12]:

```
ilorest select EthernetInterface
ilorest list Name --json
```

```
[
  {
    "Name": ""
  },
  {
    "Name": ""
  },
  {
    "Name": "Manager Dedicated Network Interface"
  },
  {
    "Name": "Manager Shared Network Interface"
  },
  {
    "Name": ""
  },
  {
    "Name": ""
  },
  {
    "Name": "Manager Virtual Network Interface"
  },
  {
    "Name": "System Ethernet Interfaces"
  },
  {
    "Name": "Manager Network Interfaces"
  }
]
```

The output of the previous command shows that Dedicated iLO Network Interface can be uniquely identified using the `Name` property. The following command displays the IPv4 configuration of the Dedicated iLO Network Interface using the `--filter` option.

[13]:
```
ilorest get IPv4Addresses --filter Name="Manager Dedicated Network Interface"
```

```
IPv4Addresses=
               SubnetMask=255.255.252.0
               AddressOrigin=DHCP
               Gateway=16.31.84.1
               Address=16.31.87.100
```

[14]:
```
ilorest get IPv4Addresses/Address --filter Name="Manager Dedicated Network␣
 ↪Interface"
```

```
IPv4Addresses=
               Address=16.31.87.100
```

Note: The `Oem/Hpe/InterfaceType` property could have been used as well.

[15]:
```
ilorest get IPv4Addresses/Address --filter Oem/Hpe/InterfaceType="Dedicated"
```

```
IPv4Addresses=
               Address=16.31.87.100
```

1.5 Use of the iLOrest debug mode

This paragraph explains how the iLOrest debug mode can help you to troubleshoot your Python, Bash or PowerShell Redfish scripts.

Imagine you have difficulties to develop a Python or Bash/cURL script modifying the system boot order in order to stop the next reboot of a server at RBSU/Bios Setup.

You can use iLOrest and its `bootorder` macro command in debug mode to understand how it perfoms this task and then reproduce it in your own programs.

The next cell prepares the environment: Remove any log file if any, login, modify the cache files.

[16]:

```
# Cleanup log file if any
rm ${LogFile} &> /dev/null
```

The help of the `bootorder` extension provides the syntax to send a `OneTimeBoot` command

[17]:

```
ilorest help bootorder | grep 'onetime'
```

```
        example: bootorder --onetimeboot=Hdd
  --onetimeboot ONETIMEBOOT
```

The `bootorder` extension with no argument returns the list of all possible `OneTimeBoot` options.

[18]:

```
ilorest bootorder
```

```


Current Persistent Boot Order:
1. HD.EmbRAID.1.5 (CentOS Linux)
2. Unknown.Unknown.200.3 (CentOS)
3. Unknown.Unknown.200.2 (Red Hat Enterprise Linux)
4. Unknown.Unknown.200.1 (grub)
5. Generic.USB.1.1 (Generic USB Boot)
6. CD.Virtual.3.1 (iLO Virtual USB 3 : iLO Virtual CD-ROM)
7. HD.EmbRAID.1.3 (Embedded RAID 1 : HPE Smart Array P408i-a SR Gen10 - 279.3
GiB, RAID1 Logical Drive 1(Target:0, Lun:0))
8. HD.EmbRAID.1.4 (Embedded RAID 1 : HPE Smart Array P408i-a SR Gen10 - Size:1
TiB Port:1I Bay:2 Box:1)

Continuous and one time boot options:
1. None
2. Cd
3. Hdd
4. Usb
5. SDCard
6. Utilities
7. Diags
```

```
8. BiosSetup
9. Pxe
10. UefiShell
11. UefiHttp
12. UefiTarget

Continuous and one time boot uefi options:
1. HD.EmbRAID.1.5 (CentOS Linux)
2. Unknown.Unknown.200.3 (CentOS)
3. Unknown.Unknown.200.2 (Red Hat Enterprise Linux)
4. Unknown.Unknown.200.1 (grub)
5. Generic.USB.1.1 (Generic USB Boot)
6. CD.Virtual.3.1 (iLO Virtual USB 3 : iLO Virtual CD-ROM)
7. HD.EmbRAID.1.3 (Embedded RAID 1 : HPE Smart Array P408i-a SR Gen10 - 279.3
GiB, RAID1 Logical Drive 1(Target:0, Lun:0))
8. HD.EmbRAID.1.4 (Embedded RAID 1 : HPE Smart Array P408i-a SR Gen10 - Size:1
TiB Port:1I Bay:2 Box:1)
```

The following cell sends the desired `bootorder` command and retrieves the list of all modifies parameters. This is interesting information but you may need more details to understand exactly how to send the request to the remoter iLO 5.

[19]:

```
ilorest bootorder --onetimeboot=BiosSetup
ilorest status
```

```
Current changes found:
ComputerSystem.v1_10_0(/redfish/v1/Systems/1/) (Currently selected)
        Boot/BootSourceOverrideTarget=BiosSetup
        Boot/BootSourceOverrideEnabled=Once
```

The following command commits the changes in **debug mode** in the remore iLO 5.

Debug messages are redirected a log file, `stdout` and `stderr`. However, `stdout` and `stderr` are discarded to `/dev/null` to keep a clean display in this noteobook.

[20]:

```
ilorest --debug commit &> /dev/null

# Cleaning manually pending changes in the cache to mimic a real iLO 5
ilorest select $(ilorest select | awk '{print $NF}') --refresh
```

1.5.1 Debug file analysis

An `iLOrest.log` file should appear soon in your left Jupyter Sidebar. Double click on it and search for string : `PATCH`

Analyze the PATCH request. Interesting for your debugging are:

- PATH / target URI: `/redfish/v1/Systems/1/`

- BODY / Payload request: `{"Boot": {"BootSourceOverrideTarget": "BiosSetup", "BootSourceOverrideEnabled": "Once"}}`
- Response Code: `Code:204 No Content`
- Response BODY: `BODY: None`

  NOTE: In this simulator environment the Response code is 204 (No Content) and thus the BODY is empty (`None`). If you send this request toward a real iLO 5, you will get a non-empty response.

1.6 File-based mode

In this section, you will study the `save/load` and `serverclone` commands. Note that the `iloclone` command is being deprecated.

To deploy / load configuration files in parallel towards multiple targets, you can read this article.

1.6.1 Save/load a specific data type configuration

In this section, you will the `save` and `load` macro iLOrest commands to save and then deploy a specific or multiple data types.

By default, the output of the save macro command is written in the `ilorest.json` file in the current directory. This can be altered by the `-f` command option.

The following cell retrieves the `Chassis`and `ManagerNetworkProtocol` data types. After its execution, you will see file `ilorest.json` appearing in your left sidebar. It can take 10-15 seconds to appear in the left sidebar.

[21]:

```
# Note: there is NO space character between list items
ilorest save --multisave Chassis,ManagerNetworkProtocol
```

```
Saving configuration...
Configuration saved to: ilorest.json
```

You can double click on the `ilorest.json` file to review its content. In addition to the `Chassis` and `ManagerNetworkProtocol` data types, it contains a `Comments` section with "administrative" properties.

The following cell modifies two parameters (`IndicatorLED` and `AlertMailEmail`) in the `ilorest.json` file.

[22]:

```
sed -i -e 's/\("IndicatorLED": \).*/\1"Blinking",/ ; s/\("AlertMailEmail": \).*/
 ↪\1"John Deuf",/' ilorest.json
```

The following cell loads the modified configuration files into its cache, displays the modifications and commits them to the managed server.

[23]:

```
echo "Loading modified properties into target system"
ilorest load

echo -e "\nPrint Status"
```

```
ilorest status

echo -e "\nCommit changes"
ilorest commit

# Clean up manually pending changes in the cache to mimic a real iLO 5
ilorest select ManagerNetworkProtocol. --refresh
ilorest select Chassis. --refresh
```

```
Loading modified properties into target system
Loading configuration...
Committing changes...
The operation completed successfully.
The operation completed successfully.

Print Status
Current changes found:
ManagerNetworkProtocol.v1_0_0(/redfish/v1/Managers/1/NetworkProtocol/)
        Oem/Hpe/AlertMailEmail=John Deuf
Chassis.v1_6_0(/redfish/v1/Chassis/1/)
        IndicatorLED=Blinking

Commit changes
Committing changes...
The operation completed successfully.
The operation completed successfully.
```

1.6.2 Save/load the entire configuration of a server

The `serverclone save` and `serverclone load` commands respectively create from a server or restore to a server a JSON formatted file containing most of its configuration settings (Smart Array configuration is not saved by default). The default output file is `ilorest_clone.json` but can be altered with the command `-f` parameter. Smart Array controller settings and logical drive configurations can be optionally included for saving.

By default the `serverclone save` command asks the user to provide inputs (i.e passwords) to be included in the output configuration file. If you don't want to supply them manually during the save operation, you can provide the `--auto` parameter like in the following cell. A placeholder will be inserted in the output file for later editing.

The following cell retrieves the configuration parameters of a remove iLO 5 based server without the SSA parameters.

[24]:
```
# Retrieve server configuration including SSA parameters but without Bios␣
 ↪parameters.
ilorest serverclone save --auto --nobios --ilossa
```

```
Bios configuration will be excluded.
Saving properties of type: AccountService, path: /redfish/v1/AccountService/
Saving properties of type: ComputerSystem, path: /redfish/v1/Systems/1/
Saving properties of type: EthernetInterface, path:
/redfish/v1/Managers/1/EthernetInterfaces/1/
Saving properties of type: EthernetInterface, path:
/redfish/v1/Managers/1/EthernetInterfaces/2/
Saving properties of type: EthernetInterface, path:
/redfish/v1/Managers/1/EthernetInterfaces/3/
Saving properties of type: HpeESKM, path:
/redfish/v1/Managers/1/SecurityService/ESKM/
An error occurred saving type: HpeServerBootSettings
Error: ""
Saving properties of type: HpeiLODateTime, path:
/redfish/v1/Managers/1/DateTime/
Remember to edit the Federation key for acct: 'DEFAULT'.
Saving properties of type: HpeiLOFederationGroup, path:
/redfish/v1/Managers/1/FederationGroups/DEFAULT/
License Key Found ending in: DDNMM
License Key Found ending in: DDNMM
Remember to verify your License Key...
Saving properties of type: HpeiLOLicense, path:
/redfish/v1/Managers/1/LicenseService/1/
Saving properties of type: HpeiLOSSO, path:
/redfish/v1/Managers/1/SecurityService/SSO/
Saving properties of type: HpeiLOSnmpService, path:
/redfish/v1/Managers/1/SnmpService/
Saving properties of type: Manager, path: /redfish/v1/Managers/1/
Remember to edit password for user: 'Administrator', login name:
'Administrator'.
Saving properties of type: ManagerAccount, path:
/redfish/v1/AccountService/Accounts/1/
Remember to edit password for user: 'student', login name: 'student'.
Saving properties of type: ManagerAccount, path:
/redfish/v1/AccountService/Accounts/3/
Remember to edit password for user: 'demopaq', login name: 'demopaq'.
Saving properties of type: ManagerAccount, path:
/redfish/v1/AccountService/Accounts/2/
Saving properties of type: ManagerNetworkProtocol, path:
/redfish/v1/Managers/1/NetworkProtocol/
Saving properties of type: SecureBoot, path: /redfish/v1/Systems/1/SecureBoot/
Saving properties of type: SmartStorageConfig, path:
/redfish/v1/systems/1/smartstorageconfig/settings/
Saving of clone file to 'ilorest_clone.json' is complete.
```

You can edit the `ilorest_clone.json` file, modify it and load it in suitable servers. The `--auto` paramter avoids manual inputs. When the entire file has been loaded, a server reset is automatically performed.

> Note: The HTTP responses of the iLO 5 simulator may not be identical to the responses of a real iLO 5.

[25]:

```
ilorest serverclone load --auto -f ilorest_clone.json

# Clean up of pending changes in the cache to mimic a real iLO 5
ilorest select SmartStorageConfig. --refresh
```

```
This system has iLO Version iLO 5 v2.30.
This system has BIOS Version U32.
BIOS Versions are compatible.
This system has has iLO 5 with firmware revision 2.30.
iLO Versions are fully compatible.
Attempting Clone from a 'ProLiant DL360 Gen10' to a 'ProLiant DL360 Gen10'.
Type '#SecureBoot.v1_0_0.SecureBoot' is compatible with this system.
Type '#HpeiLOFederationGroup.v2_0_0.HpeiLOFederationGroup' is compatible with
this system.
Type '#ComputerSystem.v1_10_0.ComputerSystem' is compatible with this system.
Type '#HpeiLODateTime.v2_0_0.HpeiLODateTime' is compatible with this system.
Type '#ManagerNetworkProtocol.v1_0_0.ManagerNetworkProtocol' is compatible with
this system.
Type '#AccountService.v1_5_0.AccountService' is compatible with this system.
Type '#EthernetInterface.v1_4_1.EthernetInterface' is compatible with this
system.
Type '#ManagerAccount.v1_3_0.ManagerAccount' is compatible with this system.
Type '#HpeiLOLicense.v2_3_0.HpeiLOLicense' is compatible with this system.
Type '#SmartStorageConfig.v2_0_1.SmartStorageConfig' is compatible with this
system.
Type '#HpeiLOSnmpService.v2_3_0.HpeiLOSnmpService' is compatible with this
system.
Type '#HpeiLOSSO.v2_0_0.HpeiLOSSO' is compatible with this system.
Type '#HpeESKM.v2_0_0.HpeESKM' is compatible with this system.
Type '#Manager.v1_5_1.Manager' is compatible with this system.
The default password will be attempted.This account already exists on this
system: 'DEFAULT'
Changing Federation account: 'DEFAULT's key
[204] The operation completed successfully.
Adding privs to Federation account: 'DEFAULT'
An invalid response body was returned: No JSON object could be decodedThe
operation completed successfully.
The default password will be attempted.The account name 'Administrator' exists
on this system. Checking for account modifications.
Changing account password for 'Administrator'.
The default password will be attempted.The account name 'student' exists on this
system. Checking for account modifications.
Changing account password for 'student'.
The default password will be attempted.The account name 'demopaq' exists on this
system. Checking for account modifications.
Changing account password for 'demopaq'.
Attempting to load a license key to the server.
```

```
The operation completed successfully.
An invalid response body was returned: No JSON object could be decodedNo error
message returned or unable to parse error response.
Patching remaining data.
Patching '#SecureBoot.v1_0_0.SecureBoot'.
Loading configuration...
No differences identified from current configuration.
Patching '#ComputerSystem.v1_10_0.ComputerSystem'.
Loading configuration...
No differences identified from current configuration.
Patching '#HpeiLODateTime.v2_0_0.HpeiLODateTime'.
Loading configuration...
No differences identified from current configuration.
Patching '#ManagerNetworkProtocol.v1_0_0.ManagerNetworkProtocol'.
Loading configuration...
No differences identified from current configuration.
Patching '#AccountService.v1_5_0.AccountService'.
Loading configuration...
No differences identified from current configuration.
Patching '#EthernetInterface.v1_4_1.EthernetInterface'.
Loading configuration...
Skipping property vlan, not found in current server.
Skipping property statelessaddressautoconfig, not found in current server.
Skipping property hostname, not found in current server.
Skipping property fqdn, not found in current server.
Skipping property dhcpv4, not found in current server.
Skipping property dhcpv6, not found in current server.
Skipping property oem, not found in current server.
Skipping property autoneg, not found in current server.
Patching '#EthernetInterface.v1_4_1.EthernetInterface'.
Loading configuration...
Committing changes...
The operation completed successfully.
The operation completed successfully.
The operation completed successfully.
The operation completed successfully.
The operation completed successfully.
Patching '#EthernetInterface.v1_4_1.EthernetInterface'.
Loading configuration...
Skipping property hostname, not found in current server.
Skipping property statelessaddressautoconfig, not found in current server.
Skipping property vlan, not found in current server.
Skipping property dhcpv4, not found in current server.
Skipping property dhcpv6, not found in current server.
Skipping property oem, not found in current server.
Patching '#SmartStorageConfig.v2_0_1.SmartStorageConfig'.
Loading configuration...
Committing changes...
```

```
The operation completed successfully.
The operation completed successfully.
The operation completed successfully.
The operation completed successfully.
The operation completed successfully.
The operation completed successfully.
Patching '#HpeiLOSnmpService.v2_3_0.HpeiLOSnmpService'.
Loading configuration...
No differences identified from current configuration.
Patching '#HpeiLOSSO.v2_0_0.HpeiLOSSO'.
Loading configuration...
No differences identified from current configuration.
Patching '#HpeESKM.v2_0_0.HpeESKM'.
Loading configuration...
No differences identified from current configuration.
Patching '#Manager.v1_5_1.Manager'.
Loading configuration...
No differences identified from current configuration.
Resetting System...

After the server is rebooted the session will be terminated.
Please wait for the server to boot completely to login again.
Rebooting server in 3 seconds...
The operation completed successfully.
Resetting iLO...

After iLO resets the session will be terminated.
Please wait for iLO to initialize completely before logging in again.
This process may take up to 3 minutes.

An error occured during iLO reset.
```

1.7 Raw commands

iLOrest allows you to get and set parameters directly in the Redfish tree using the following “raw” commands: `rawdelete`, `rawget`, `rawput`, `rawpost`, `rawhead` and `rawpatch`.

The use of raw commands is not recommended as it assumes the resource URIs. Resource URIs can change over time as explained in this article. However, in some specific cases or for troubleshooting they can be useful.

1.7.1 Raw get

The `rawget` command fetches the content of the supplied URI. Refer to the API Reference Document to find the location of your desired URI. Additional tools have to be used to filter specific properties.

The following cell intends to retrieve the IPv4 address of the iLO Dedicated Network port, assuming it is the first NIC in the collection. It uses the popular JsonQuery (`jq`) utility to filter desired parameters.

[26]:

```
# Get the Name and the IPv4 address parameters of the first iLO NIC.
ilorest rawget "/redfish/v1/Managers/1/EthernetInterfaces/1" | jq '.Name, .
 ↪IPv4Addresses[].Address'
```

```
The operation completed successfully.
"Manager Dedicated Network Interface"
"16.31.87.100"
```

1.7.2 Raw patch, put and post

The `rawpatch`, `rawput` and `rawpost` commands require a target location URI and a "body/workload" companion json file containing the parameters to patch, put or post.

> **NOTE**: The body/workload file format has changed between iLOrest versions 2.X and 3. Issue `ilorest help rawpatch` to get the file format suitable for your iLOrest version.

The following cell creates a .json file containing the body of a patch request asking the next reboot to stop at RBSU. The content of this file comes from the debug paragraph studied earlier.

Then, it executes a `rawpatch` command of this file as input. As we are using an iLO 5 simulator, the response message may not be the one received by a real physical iLO 5.

> Note: The `ilorest status` command returns "No changes found" because the `rawpatch` command is not cached.

[27]:

```
cat > RawPatchFile.json << __EOF__
{
   "/redfish/v1/Systems/1":
   {
     "Boot": {
             "BootSourceOverrideTarget": "Utilities",
             "BootSourceOverrideEnabled": "Once"
             }
   }
}
__EOF__

# Send the PATCH workload
ilorest rawpatch RawPatchFile.json

# rawpatch bypasses the cache files. Hence ilorest will not detect any changes.
ilorest status

# Logout
ilorest logout
```

```
The operation completed successfully.
Current changes found:
```

```
EthernetInterface.v1_4_1(/redfish/v1/Systems/1/EthernetInterfaces/2/)
        InterfaceEnabled=False
EthernetInterface.v1_4_1(/redfish/v1/Systems/1/EthernetInterfaces/3/)
        InterfaceEnabled=False
EthernetInterface.v1_4_1(/redfish/v1/Managers/1/EthernetInterfaces/1/)
        InterfaceEnabled=False
EthernetInterface.v1_4_1(/redfish/v1/Systems/1/EthernetInterfaces/4/)
        InterfaceEnabled=False
EthernetInterface.v1_4_1(/redfish/v1/Systems/1/EthernetInterfaces/1/)
        InterfaceEnabled=False
Logging session out.
```

1.8 Summary

In this workshop, you discovered the HPE iLOrest command line interface and its three operational modes: interactive, scripted and file based. More examples can be studied in the next Jupyter Notebook.