F-Response API Manual

8.7.1.33

Provides a complete breakdown of leveraging F-Response's API to perform expert remote e-discovery, computer forensics, and incident response.

Contents

Terminology

The following terminology will be used throughout this manual.

EXAMINER

F-Response Examiner refers to the applications used to connect to remote Subjects and Providers to attach devices and shares.

SUBJECT

F-Response Subject refers to the applications used to present remote devices, drives, memory and shares to Examiners as defined above.

PROVIDER

Provider refers to the supported 3rd party Cloud Services providers that F-Response is able to connect to and present data from.

TARGET

Targets refer to individual devices, shares, and data sources presented by Subjects or Providers to Examiners as defined above.

PHYSICAL DEVICE

Physical Device refers to the F-Response connected subject's remote physical disks and logical volumes presented as locally attached physical disks.

PHYSICAL IMAGE

Physical Image refers to an Expert Witness (EWF) formatted full device acquisition. Physical Images will include the allocated and unallocated content of the physical device. Physical images can only be performed against Physical Devices.

Overview

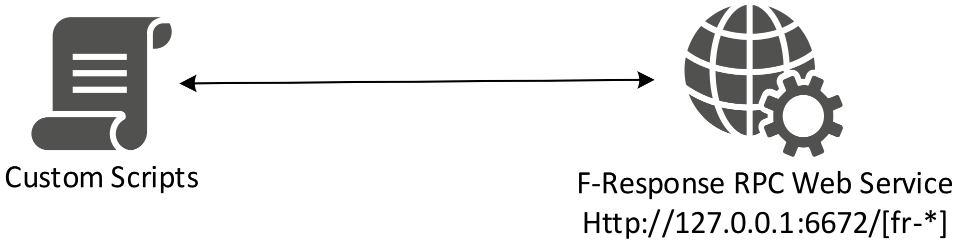
F-Response is a software product which leverages our patented and patent pending technology to provide access to remote drives, memory, volumes, and 3rd party cloud storage. This document outlines the scriptable application programming interface (API) F-Response v8 offers.

Scripting Requirements

Scripting F-Response requires a programming language and environment that can handle the following:

- HTTP POST
- JSON (Read and Write)

F-Response Scripting Model



F-Response uses JSON POST style communication between the script and a localhost bound webservice on the examiner machine to perform commands and receive responses.

F-Response Scripting Endpoints

The following are the current URI endpoints and what they provide:

- /fr-image
  - Interface for creating physical images of remote F-Response Targets.
- /fr-logic
  - Interface for deploying and connecting to remote subjects, F-Response CE, CE+C, and EE only.
- /fr-univ
  - Interface for deploying and connecting to remote subjects, F-Response Universal only.

JSON Request Format

F-Response JSON POST requests must be in the following format to be understood by the F-Response RPC Web Service.

```
{
        "function": "list_api",
        "values": {
                "value1": "one",
                "value2": 2
        }
}
```

- Function (“function”)
  - This string value represents the function your script wishes to call. You will find a complete list of the available functions by endpoint further in this document.
- Values (“values”)
  - This object contains zero or more values depending on the chosen function.

Example Request

```
POST /fr-logic HTTP/1.0
Content-Type: application/json
…

{
        "function": "list_api",
        "values": {}
}
```

JSON Response Format

F-Response JSON POST responses are in JSON format and conform to one of the two following formats:

200 – OK Responses

```
{
        "function": "list_api",
        "response": {}
}
```

A successful response includes the function name and a response object with contents specific to the request.

400 – ERROR Responses

```
{
        "error": "error text"
}
```

A failure response includes error text that contains additional details on the unsuccessful request.

Function Reference

F-Response Logic Service (CE, CE+C, EE) Endpoint - /fr-logic

get_license_manager_state

Values: None.

Help: Returns the current license manager state, stopped or running.

Example:

```
POST /fr-logic
…
{
 “function”:”get_license_manager_state”,
 “values”:{}
}

200 OK

{
  "function": "get_license_manager_state",
  "response": {
    "state": "running"
  }
}
```

get_deployment_credentials

Values: None

Help: Returns a JSON of all deployment credentials.

Example:

```
POST /fr-logic
…
{
 “function”:”get_deployment_credentials”,
 “values”:{}
}

200 OK
{
  "function": "get_deployment_credentials",
  "response": {
    "unix_credentials": [
      {
        "credentialid": "1a0464e1-e512-49c8-8b7f-edf32bf0e0b2",
        "username": "test",
        "password": "********",
        "rtpassword": "********",
        "key": "",
        "atype": 0,
        "utype": 0,
        "ptype": 0
      }
    ],
    "windows_credentials": [
      {
        "credentialid": "111fecc8-9609-44a6-8fe4-d376cdf1468c",
        "username": "frestest",
        "password": "********",
        "domain": "testdomain"
      }
    ]
  }
}
```

export_msi_options

Values: None

Help: Returns a JSON of the MSI Export options.

Example:

```
POST /fr-logic
…
{
  “function”:”export_msi_options”,
  “values”:{}
}

200 OK

{
  "function": "export_msi_options",
  "response": {
    "servicename": "F-Response Subject v81",
    "servicedesc": "F-Response Subject Service",
    "exename": "subject_srv.exe",
    "lmuri": "192.168.1.9:5682",
    "port": 3262,
    "autostart": 1,
    "addresses": [
      "::1",
      "::1",
      "192.168.1.9",
      "x64-win8-dev.fresponse.local"
    ],
    "lmport": "5682"
  }
}
```

export_gui_options

Values: None

Help: Returns a JSON of the GUI Export options.

Example:

```
POST /fr-logic
…
{
  “function”:”export_gui_options”,
  “values”:{}
}

200 OK

{
  "function": "export_gui_options",
  "response": {
    "servicename": "F-Response Subject v81",
    "servicedesc": "F-Response Subject Service",
    "exename": "subject_srv.exe",
    "lmuri": "192.168.1.9:5682",
    "port": 3262,
    "autostart": 1,
    "addresses": [
      "::1",
      "::1",
      "192.168.1.9",
      "x64-win8-dev.fresponse.local"
    ],
    "lmport": "5682"
  }
}
```

set_license_manager_details

Values: username,password,autostart,port

Help: Sets the license manager settings.

Example:

```
POST /fr-logic
…
{
  “function”:”set_license_manager_details”,
  “values”:{
               “port”:”5682”,
               “autostart”:1
       }
}

200 OK
{
  "function": "set_license_manager_details",
  "response": {
    "port": "5682",
    "threadcount": 50,
    "autostart": 1
  }
}
```

get_license_manager_details

Values: None

Help: Returns a JSON object of current license manager settings.

Example:

```
POST /fr-logic
…
{
  “function”:”get_license_manager_details”,
  “values”:{}
}

200 OK
{
  "function": "get_license_manager_details",
  "response": {
    "port": "5682",
    "threadcount": 50,
    "autostart": 1
  }
}
```

stop_license_manager

Values: None

Help: Stops the license manager.

Example:

```
POST /fr-logic
…
{
  “function”:”stop_license_manager”,
  “values”:[]
}

200 OK
{
  "function": "stop_license_manager",
  "response": {
    "lm": "status",
    "status": "stopped"
  }
}
```

start_license_manager

Values: None

Help: Starts the license manager.

Example:

```
POST /fr-logic
…
{
  “function”:”start_license_manager”,
  “values”:[]
}

200 OK
{
  "function": "start_license_manager",
  "response": {
    "lm": "status",
    "status": "running"
  }
}
```

get_license_details

Values: None

Help: Returns a JSON object of the license details for the locally attached dongle.

Example:

```
POST /fr-logic
…
{
  “function”:”get_license_details”,
  “values”:{}
}

200 OK
{
  "function": "get_license_details",
  "response": {
    "hwid": "155521729",
    "expires": "Mon, 26 Aug 2019 05:00:00 GMT",
    "type": "Enterprise",
    "type_id": 3
  }
}
```

get_license_managed_subjects

Values: None

Help: Returns a JSON array of subjects currently talking to the license manager.

Example:

```
POST /fr-logic
…
{
  “function”:”get_license_managed_subjects”,
  “values”:{}
}

200 OK
{
  "function": "get_license_managed_subjects",
  "response": {
    "subjects": [
      {
        "subject": {
          "hostname": "x64-win2k12-sub",
          "subjecturi": "192.168.1.219:3262/sub",
          "platform": "Windows Server 2012",
          "version": "8.0.0.52",
          "uuid": "b09ff0a5-8e1f-43b8-a390-f55440dec34d",
          "type": 1
        }
      }
    ]
  }
}
```

undeploy

Values: subject, (optional: credentialid)

Help: Undeploys the subject software from a remote machine.

Example:

```
POST /fr-logic
{
       “function”:”undeploy”,
       “values”:{
               “subject”:”x64-win2k12-sub”,
**Optional** “credentialid": "1a0464e1-e512-49c8-8b7f-edf32bf0e0b2"

       }
}

200 OK
{
  "function": "undeploy",
  "response": {
    "subject": "x64-win2k12-sub",
    "state": "stopped"
  }
}
```

deploy

Values: subject (optional: credentialid)

Help: Deploys and starts the subject software on a remote machine.

Example:

```
POST /fr-logic
{
       “function”:”deploy”,
       “values”:{
               “subject”:”x64-win2k12-sub”,
**Optional** “credentialid": "1a0464e1-e512-49c8-8b7f-edf32bf0e0b2"
       }
}

200 OK
{
  "function": "deploy",
  "response": {
    "subject": "x64-win2k12-sub",
    "state": "running"
  }
}
```

add_unix_deployment_credential

Values: username, password, rtpassword, key, assume

Help: Adds a unix deployment credential.

Example:

```
POST /fr-logic
{
       “function”:”add_unix_deployment_credential”,
       “values”:{
               “username”:”root or username”,
               “password”:”password”,
               “rtpassword”:”su password if assuming root using su”,
               “key”:”ssh key, optional, leave blank if not needed.”,
               “assume”: “su, sudo, or leave blank if not needed.”
       }
}

200 OK
{
  "function": "add_unix_deployment_credential",
  "response": {
    "unix_credentials": [
      {
        "credentialid": "1a0464e1-e512-49c8-8b7f-edf32bf0e0b2",
        "username": "test",
        "password": "********",
        "rtpassword": "********",
        "key": "",
        "atype": 0,
        "utype": 0,
        "ptype": 0
      }
    ]
  }
}
```

add_windows_deployment_credential

Values: username, domain, password

Help: Adds a windows deployment credential.

Example:

```
POST /fr-logic
{
        "function": "add_windows_deployment_credential",
        "values": {
                "username": "frestest",
                "domain": "testdomain",
                "password": "password123"
        }
}
200 OK
{
  "function": "add_windows_deployment_credential",
  "response": {
    "windows_credentials": [
      {
        "credentialid": "111fecc8-9609-44a6-8fe4-d376cdf1468c",
        "username": "frestest",
        "password": "********",
        "domain": "testdomain"
      }
    ]
  }
}
```

remove_unix_deployment_credential

Values: credentialid

Help: Remove a unix deployment credential.

Example:

```
POST /fr-logic
{
        "function": "remove_unix_deployment_credential",
        "values": {
                “credentialid”:”111fecc8-9609-44a6-8fe4-d376cdf1468c”
        }
}
200 OK
{
  "function": "remove_unix_deployment_credential",
  "response": {
    "unix_credentials": []

  }
```

remove_windows_deployment_credential

Values: credentialid

Help: Remove a windows deployment credential.

Example:

```
POST /fr-logic
{
       "function": "remove_windows_deployment_credential",
       "values": {
              “credentialid”:”111fecc8-9609-44a6-8fe4-d376cdf1468c”
       }
}
200 OK
{
  "function": "remove_windows_deployment_credential",
  "response": {
    "windows_credentials": []

  }
```

export_gui

Values: platform, lmuri, exportpath

Help: Exports a GUI executable.

Example:

```
POST /fr-logic
…
{
  “function”:”export_gui”,
  “values”:{
       “platform”:” sub-aix-power or sub-osx-x64 or sub-lin-i386 or sub-lin-x64 or sub-sun-i386
or sub-sun-sparc or sub-win-i386" or sub-win-x64"
       “lmuri”:”ADDRESS:PORT of your license manager, see get_available_addresses for a list”,
       “exportpath”:”Local destination, may not be a network share, example c:\\”
  }
}

200 OK
{
  "function": "export_gui",
  "response": {
    "exportedgui": "c:\\\\sub-win-x64_192.168.1.1_5682_3262.exe"
  }
}
```

export_msi

Values: servicename, servicedesc, exename, port, lmuri, autostart, exportpath

Help: Exports an MSI.

Example:

```
POST /fr-logic
…
{
  “function”:”export_msi”,
  “values”:{
       “servicename”:”F-Response v8 Service",
       “servicedesc”:”F-Response Service Description”,
       “exename”:”f-response-srv.exe”,
       “port”:3262
       “lmuri”:”ADDRESS:PORT of your license manager, see get_available_addresses for a list”,
       “exportpath”:”Local destination, may not be a network share, example c:\\”
  }
}

200 OK
{
  "function": "export_msi",
  "response": {
    "exportedgui": "c:\\\\f-response.msi"
  }
}
```

unmount_target

Values: workeruuid,

Help: Unmounts a subject device by workeruuid

Example:

```
POST /fr-logic
…
{
   “function”:”unmount_target”,
   “values”:{
        “workeruuid”:”105b49f9-6436-41b9-8c8d-4fa758e86eb1”
        }
}

200 OK
{
  "function": "unmount_target",
  "response": {
    "workers": []
  }
}
```

mount_target

Values: subjecturi, targetid

Help: Mounts a subject device by subjecturi and targetid.

Example:

```
POST /fr-logic
…
{
  “function”:”mount_target”,
  “values”:{
        “subjecturi”:”192.168.1.219:3262/sub”,
        “targetid”:0
        }
}

200 OK
{
  "function": "mount_target",
  "response": {
    "workeruuid": "105b49f9-6436-41b9-8c8d-4fa758e86eb1",
    "subjecturi": "192.168.1.219:3262/sub",
    "subjectname": "192.168.1.219:3262/sub",
    "targetname": "0",
    "targetid": "0",
    "state": "pending",
    "drive": "attaching...",
    "targetdriveid": "2399B26F5C67B92C",
  }
}
```

stop_remote_subject

Values: subjecturi

Help: Stop and remove F-Response from a single remote subject by subjecturi.

Example:

```
POST /fr-logic
…
```

```
{
  “function”:”stop_remote_subject”,
  “values”:{
        “subjecturi”:”192.168.1.219:3262/sub”
        }
}

200 OK
{
  "function": "stop_remote_subject",
  "response": {
    "subjecturi": "192.168.1.219:3262/sub",
    "state": "stopped"
  }
}
```

get_mounted_targets

Values: None

Help: Returns a JSON of all mounted devices.

Example:

```
POST /fr-logic
…
{
  “function”:”get_mounted_targets”,
  “values”:{}
}

200 OK
{
  "function": "get_mounted_targets",
  "response": {
    "workers": [
      {
        "workeruuid": "105b49f9-6436-41b9-8c8d-4fa758e86eb1",
        "subjecturi": "192.168.1.219:3262/sub",
        "subjectname": "x64-win2k12-sub",
        "targetname": "disk-0",
        "targetid": "0",
        "state": "connected",
        "drive": "\\\\.\\PhysicalDrive2",
        "targetdriveid": "2399B26F5C67B92C",
        "blocksize": "512",
        "blockcount": "83886080"
      }
    ]
  }
}
```

set_deployment_details

Values: servicename, servicedesc, exename, port, lmuri, autostart

Help: Sets one or more deployment settings.

Example:

```
POST /fr-logic
…
{
  “function”:”set_deployment_details”,
  “values”:{
        "servicename": "F-Response Subject v81",
        "servicedesc": "F-Response Subject Service",
        "exename": "subject_srv.exe",
        "lmuri": "192.168.1.9:5682",
        "port": 3262,
        "autostart": 1
        }
}

200 OK
{
  "function": "get_deployment_details",
  "response": {
    "servicename": "F-Response Subject v81",
    "servicedesc": "F-Response Subject Service",
    "exename": "subject_srv.exe",
    "lmuri": "192.168.1.9:5682",
    "port": 3262,
    "autostart": 1,
    "addresses": [
      "::1",
      "::1",
      "192.168.1.9",
      "x64-win8-dev.fresponse.local"
    ],
    "lmport": "5682"
  }
}
```

get_deployment_details

Values: None

Help: Returns a JSON object of all the deployment options.

Example:

```
POST /fr-logic
…
{
  “function”:”get_deployment_details”,
  “values”:{}
}

200 OK
{
  "function": "get_deployment_details",
  "response": {
    "servicename": "F-Response Subject v81",
    "servicedesc": "F-Response Subject Service",
    "exename": "subject_srv.exe",
    "lmuri": "192.168.1.9:5682",
    "port": 3262,
    "autostart": 1,
    "addresses": [
      "::1",
      "::1",
      "192.168.1.9",
      "x64-win8-dev.fresponse.local"
    ],
    "lmport": "5682"
  }
}
```

get_subject_targets

Values: subjecturi

Help: Returns a list of available targets for a given subjecturi.

Example:

```
POST /fr-logic
…
{
  “function”:”get_subject_targets”,
  “values”:{
       “subjecturi”:”192.168.1.219:3262/sub”
       }
}

200 OK
{
  "function": "get_subject_targets",
  "response": {
    "subjectname": "x64-win2k12-sub",
    "subjecturi": "192.168.1.219:3262/sub",
    "uuid": "a7f78b4e-c573-4d94-9684-39957150fdb5",
    "platform": "Windows Server 2012",
    "version": "8.0.0.52",
    "type": 1,
    "targets": [
      {
        "target": {
          "name": "pmem-unsafe",
          "id": "4189",
          "type": "3",
          "blocksize": "4096",
          "blockcount": "1310720"
        }
      },
      {
        "target": {
          "name": "pmem",
          "id": "4187",
          "type": "3",
          "blocksize": "4096",
          "blockcount": "1310720"
        }
      },
      {
        "target": {
          "name": "vol-E",
          "id": "4165",
          "type": "2",
          "blocksize": "512",
          "blockcount": "15474816"
        }
      },
      {
        "target": {
          "name": "vol-C",
          "id": "4163",
          "type": "2",
          "blocksize": "512",
          "blockcount": "83165184"
        }
      },
      {
        "target": {
          "name": "disk-2",
          "id": "2",
          "type": "1",
          "blocksize": "512",
          "blockcount": "15482880"
        }
      },
      {
        "target": {
          "name": "disk-1",
          "id": "1",
          "type": "1",
          "blocksize": "512",
          "blockcount": "10485760"
        }
      },
      {
        "target": {
          "name": "disk-0",
          "id": "0",
          "type": "1",
          "blocksize": "512",
          "blockcount": "83886080"
        }
      }
    ]
  }
}
```

list_api

Values: None

Help: Lists the full API for this endpoint.

F-Response Imaging Service (CE, CE+C, EE) Endpoint - /fr-image

stop_image

Values: workeruuid

Help: Stops a running image operation.

Example:

```
POST /fr-image
…
{
        “function”:”stop_image”,
        “values”:{
                “workeruuid”:” 7233a4e9-785a-49c3-abda-fdca1192fc94”
        }
}

200 OK
{
  "function": "stop_image",
  "response": {
    "workers": [
      {
        "imager": {
          "image_name": "test-disk1",
          "image_path": "c:\\collection\\",
          "hash": "md5"
        },
        "ewf": {
          "case_number": "not_provided",
          "evidence_number": "not provided",
          "examiner_name": "testexaminer",
          "notes": "testnotes",
          "unique_description": "testdesc"
        },
        "fresponse": {
          "subject": "192.168.1.219:3262/sub",
          "target": "disk-1",
        },
        "workeruuid": "7233a4e9-785a-49c3-abda-fdca1192fc94",
        "state": "stopped",
        "eventid": "98880D0F0B85C543",
        "perf": "116 MB/s - 100%"
      }
    ]
  }
}
```

remove_image

Values: workeruuid

Help: Removes an image by workeruuid.

Example:

```
POST /fr-image
…
{
        “function”:”remove_image”,
        “values”:{
                “workeruuid”:” 7233a4e9-785a-49c3-abda-fdca1192fc94”
        }
}

200 OK
```

```
{
  "function": "remove_image",
  "response": {
    "workers": [
    ]
  }
}
```

list_images

Values: None

Help: Returns a JSON array of all running images.

Example:

```
POST /fr-image
…
{
       “function”:”list_images”,
       “values”:{}
}

200 OK
{
  "function": "list_images",
  "response": {
    "workers": [
      {
        "imager": {
          "image_name": "test-disk1",
          "image_path": "c:\\collection\\",
          "hash": "md5"
        },
        "ewf": {
          "case_number": "not_provided",
          "evidence_number": "not provided",
          "examiner_name": "testexaminer",
          "notes": "testnotes",
          "unique_description": "testdesc"
        },
        "fresponse": {
          "subject": "192.168.1.219:3262/sub",
          "target": "disk-1",
        },
        "workeruuid": "7233a4e9-785a-49c3-abda-fdca1192fc94",
        "state": "completed",
        "eventid": "98880D0F0B85C543",
        "perf": "116 MB/s - 100%"
      }
    ]
  }
}
```

start_physical_image

Values: image_name, image_path, hash, source

Help: Starts a physical image.

Example:

```
POST /fr-image
…
There are two different imaging options. Please refer to the following JSON request formats.

For creating an expert witness format image of an attached disk when you have a dongle inserted
in your examiner machine.
```

```
{
        "function":"start_physical_image",
        "values":{
                "image_name": "test-disk1",
                "image_path":"c:\\collection\\",
                "hash":"md5",
                “case_number":"case1",
                "evidence_number":"ev1",
                "examiner_name":"testexaminer",
                "notes":"testnotes",
                "source":"disk",
                "path":"\\.\PhysicalDrive2",
                "unique_description":"testdesc"
        }
}
```

For creating an expert witness format image of an attached disk when you are using a remote license manager

```
{
        "function":"start_physical_image",
        "values":{
                "image_name": "test-disk1",
                "image_path":"c:\\collection\\",
                "hash":"md5",
                “case_number":"case1",
                "evidence_number":"ev1",
                "examiner_name":"testexaminer",
                "notes":"testnotes",
                "source":"disk",
                "path":"\\.\PhysicalDrive2",
                “lmuri”:”192.168.1.1:5682/lm”,
                "unique_description":"testdesc"
        }
}
```

For creating an expert witness format image of a remote F-Response target when you have a dongle inserted in your examiner machine.

```
{
        "function":"start_physical_image",
        "values":{
                "image_name": "test-disk1",
                "image_path":"c:\\collection\\",
                "hash":"md5",
                “case_number":"case1",
                "evidence_number":"ev1",
                "examiner_name":"testexaminer",
                "notes":"testnotes",
                "subjecturi":"192.168.1.219:3262/sub",
                "targetname":"disk-1",
                “source”:”fresponse”,
                "unique_description":"testdesc"
        }
}
```

For creating an expert witness format image of a remote F-Response target when you are using a remote license manager.

```
{
        "function":"start_physical_image",
        "values":{
                "image_name": "test-disk1",
                "image_path":"c:\\collection\\",
                "hash":"md5",
                “case_number":"case1",
                "evidence_number":"ev1",
                "examiner_name":"testexaminer",
                "notes":"testnotes",
                "subjecturi":"192.168.1.219:3262/sub",
                "targetname":"disk-1",
                “source”:”fresponse”,
                “lmuri”:”192.168.1.1:5682/lm”,
                "unique_description":"testdesc"
        }
}


200 OK

{
  "function": "start_physical_image",
  "response": {
    "workers": [
      {
        "imager": {
          "image_name": "test-disk1",
          "image_path": "c:\\collection\\",
          "hash": "md5"
        },
        "ewf": {
          "case_number": "not_provided",
          "evidence_number": "not provided",
          "examiner_name": "testexaminer",
          "notes": "testnotes",
          "unique_description": "testdesc"
        },
        "fresponse": {
          "subject": "192.168.1.219:3262/sub",
          "target": "disk-1",
        },
        "workeruuid": "7233a4e9-785a-49c3-abda-fdca1192fc94",
        "state": "pending",
        "eventid": "98880D0F0B85C543"
      }
    ]
  }
}
```

start_physical_image (universal)

Values: image_name, image_path, hash, source, universal, subjectname, targetname

Help: Starts a physical image.

Example:

```
POST /fr-image
…
There are two different imaging options. Please refer to the following JSON request formats.

For creating an expert witness format image of an attached disk.

{
        "function":"start_physical_image",
        "values":{
                "image_name": "test-disk1",
                "image_path":"c:\\collection\\",
                "hash":"md5",
                “case_number":"case1",
                "evidence_number":"ev1",
                "examiner_name":"testexaminer",
                "notes":"testnotes",
                "source":"disk",
                "path":"\\.\PhysicalDrive2",
                "unique_description":"testdesc"
        }
}

For creating an expert witness format image of a remote Universal subject directly, without
attaching a disk
```

```
{
        "function":"start_physical_image",
        "values":{
                "image_name": "test-disk1",
                "image_path":"c:\\collection\\",
                "hash":"md5",
                “case_number":"case1",
                "evidence_number":"ev1",
                "examiner_name":"testexaminer",
                "notes":"testnotes",
                "source":"universal",
                "universal":"192.168.1.219",
                “subjectname”:”x64-win2k12-sub”,
                 “targetname”:”disk-0”,
                "unique_description":"testdesc"
        }
}

200 OK

{
  "function": "start_physical_image",
  "response": {
    "workers": [
      {
        "imager": {
          "image_name": "test-disk1",
          "image_path": "c:\\collection\\",
          "hash": "md5"
        },
        "ewf": {
          "case_number": "not_provided",
          "evidence_number": "not provided",
          "examiner_name": "testexaminer",
          "notes": "testnotes",
          "unique_description": "testdesc"
        },
        "universal": {
          "subjectname": "x64-win2k12-sub",
          "targetname": "disk-0",
        },
        "workeruuid": "7233a4e9-785a-49c3-abda-fdca1192fc94",
        "state": "pending",
        "eventid": "98880D0F0B85C543"
      }
    ]
  }
}
```

F-Response Universal Service (UNIV) Endpoint - /fr-univ

list_api

Values:

Help: Lists the full API for this library.

connect

Values: universal

Help: Connects to an F-Response Universal server.

Example:

```
POST /fr-univ
…
{
        “function”:”connect”,
        “values”:{
                “universal”:”192.168.0.1”
        }
}

200 OK
{
  "function": "connect",
  "response": {
    "universal": "192.168.0.1",
    "status": "connected"
  }
}
```

disconnect

Values: universal

Help: Disconnects from an F-Response Universal server.

Example:

```
POST /fr-univ
…
{
        “function”:”disconnect”,
        “values”:{
                “universal”:”192.168.0.1”
        }
}

200 OK
{
  "function": "disconnect",
  "response": {
    "universal": "192.168.0.1",
    "status": "disconnected"
  }
}
```

get_deployment_details

Values:

Help: Returns a JSON object of all the deployment options.

Example:

```
POST /fr-univ
…
{
        “function”:”get_deployment_details”,
        “values”:{
                “universal”:”192.168.0.1”
        }
}

200 OK
{
  "function": "get_deployment_details",
  "response": {
    "universals": [
      {
        "universal": "192.168.0.1",
        "adminport": 443,
        "port": 80,
        "username": "frestest",
        "autostart_subject": 0,
        "servicename": "F-Response Universal",
        "servicedesc": "F-Response Universal Service",
        "exename": "f-response-univ-srv.exe",
        "waithint": 3600
      }
    ]
  }
}
```

set_deployment_details

Values: universal, servicename, servicedesc, exename, waithint, autostart_subject

Help: Sets one or more deployment settings.

Example:

```
POST /fr-univ
…
{
        “function”:”set_deployment_details”,
        “values”:{
                “universal”:”192.168.0.1”,
                “servicename”:”F-Response Universal Service”
        }
}

200 OK
{
  "function": "get_deployment_details",
  "response": {
    "universals": [
      {
        "universal": "192.168.0.1",
        "adminport": 443,
        "port": 80,
        "username": "frestest",
        "autostart_subject": 0,
        "servicename": "F-Response Universal Service",
        "servicedesc": "F-Response Universal Service",
        "exename": "f-response-univ-srv.exe",
        "waithint": 3600
      }
    ]
  }
}
```

add_universal

Values: universal, username, password, adminport, port, autostart (optional)

Help: Adds an F-Response Universal server.

Example:

```
POST /fr-univ
…
{
        “function”:”add_universal”,
        “values”:{
                “universal”:”192.168.0.1”,
                “username”:”frestest”,
                “password”:”testpassword”,
                “adminport”:443,
                “port”:80,
               “autostart”:0
        }
}

200 OK
{
  "function": "add_universal",
  "response": {
    "universals": [
      {
         "universal": "192.168.0.1",
        "adminport": 443,
        "port": 80,
        "username": "frestest",
        "connected": 0,
        "autoconnect": 0
      }    ]
  }
}
```

remove_universal

Values: universal

Help: Removes an F-Response Universal server.

Example:

```
POST /fr-univ
…
{
        “function”:”remove_universal”,
        “values”:{
                “universal”:”192.168.0.1”
        }
}

200 OK
{
  "function": "remove_universal",
  "response": {
    "universals": []
  }
}
```

edit_universal

Values: universal, username, password,adminport,port

Help: Edits an existing F-Response Universal server.

Example:

```
POST /fr-univ
…
{
       “function”:”edit_universal”,
       “values”:{
              “universal”:”192.168.0.1”,
              “username”:”frestest”,
              “password”:”testpassword”,
              “adminport”:443,
              “port”:80,
             “autostart”:1
       }
}

200 OK
{
  "function": "edit_universal",
  "response": {
    "universals": [
      {
        "universal": "192.168.0.1",
        "adminport": 443,
        "port": 80,
        "username": "frestest",
        "connected": 1,
        "autoconnect": 1
      }    ]
  }
}
```

get_mounted_targets

Values:

Help: Returns a JSON of all mounted devices.

Example:

```
POST /fr-univ
…
{
  “function”:”get_mounted_targets”,
  “values”:{}
}

200 OK
{
  "function": "get_mounted_targets",
  "response": {
    "workers": [
      {
        "targetname": "Disk-0-40960mb",
        "subjectname": "x64-win2k12-sub",
        "uuid": 7,
        "block_count": "83886080",
        "block_size": 512,
        "type": 3,
        "case_sensitive": 0,
        "root_node": -842150451,
        "authtype": 1,
        "workeruuid": "f672b964-2bbb-4774-87f8-2b5672fc74ad",
        "universal": "192.168.0.1",
        "subjectid": "15369451274081149164",
        "state": "connected",
        "drive": "\\\\.\\PhysicalDrive4",
        "driveletter": "",
        "targetdriveid": "DA3BE4406A273111"
      }
    ]
  }
}
```

stop_remote_subject

Values: universal, subjectid

Help: Stop and remove F-Response from a single remote subject.

Example:

```
POST /fr-univ
…
{
   “function”:”stop_remote_subject”,
   “values”:{
       “universal”:”192.168.0.1”,
        “subjectid”:” 15369451274081149164”
        }
}

200 OK
{
  "function": "stop_remote_subject",
  "response": {
    "subject": {
      "id": "15369451274081149164",
      "name": "x64-win2k12-sub",
      "op": "remove",
      "universal": "192.168.0.1"
    }
  }
}
```

mount_target

Values: universal, subjectid, uuid, driveletter

Help: Mounts a subject device by universal, subjectid, uuid, and driveletter, if drive letter isn’t needed leave it blank.

Example:

```
POST /fr-univ
…
{
   “function”:”mount_target”,
   “values”:{
       “universal”:”192.168.0.1”,
        “subjectid”:” 15369451274081149164”,
        “uuid”:7,
        “driveletter”:””
        }
}

200 OK
{
  "function": "mount_target",
  "response": {
    "targetname": "Disk-0-40960mb",
    "subjectname": "x64-win2k12-sub",
    "uuid": 7,
    "block_count": "83886080",
    "block_size": 512,
    "type": 3,
    "case_sensitive": 0,
    "root_node": -842150451,
    "authtype": 1,
    "workeruuid": "b1b60fd7-1584-4f09-94ac-aae941d29f6f",
    "universal": "192.168.0.1",
    "subjectid": "15369451274081149164",
    "state": "pending",
    "drive": "attaching...",
    "driveletter": "",
    "targetdriveid": "3AA87CC05179FB13"
  }
}
```

unmount_target

Values: workeruuid

Help: Unmounts a subject device by workeruuid.

Example:

```
POST /fr-univ
…
{
   “function”:”unmount_target”,
   “values”:{
        “workeruuid”:”105b49f9-6436-41b9-8c8d-4fa758e86eb1”
        }
}

200 OK
{
  "function": "unmount_target",
  "response": {
    "workers": []
  }
}
```

export_msi

Values: servicename, servicedesc, exename, waithint, universal, exportpath

Help: Exports an MSI.

Example:

```
POST /fr-univ
…
{
   “function”:”export_msi”,
   “values”:{
        “servicename”:”F-Response Universal Service”,
        “servicedesc”:”F-Response Universal Service Description”,
        “exename”:”fresuniv.exe”,
        “waithint”:3600
        “universal”:”192.168.0.1”,
        “exportpath”:”c:\\”
        }
}

200 OK
{
  "function": "export_msi",
  "response": {
    "msi": “c:\\f-response…”
  }
}
```

export_gui

Values: platform, universal, exportpath

Help: Exports a GUI executable.

Example:

```
POST /fr-univ
…
{
   “function”:”export_gui”,
   “values”:{
        “platform”:”Windows GUI”,
       “universal”:”192.168.0.1”,
       “exportpath”:”c:\\”
        }
}

200 OK
{
  "function": "export_gui",
  "response": {
    "exportedgui": “c:\\f-response…”
  }
}
```

remove_windows_deployment_credential

Values: credentialid

Help: Removes a Windows deployment credential.

Example:

```
POST /fr-univ
{
        "function": "remove_windows_deployment_credential",
        "values": {
                “credentialid”:”111fecc8-9609-44a6-8fe4-d376cdf1468c”
        }
}
200 OK
{
  "function": "remove_windows_deployment_credential",
  "response": {
    “windows_credentials": []

  }
```

remove_unix_deployment_credential

Values: credentialid

Help: Remove a unix deployment credential.

Example:

```
POST /fr-univ
{
        "function": "remove_unix_deployment_credential",
        "values": {
                “credentialid”:”111fecc8-9609-44a6-8fe4-d376cdf1468c”
        }
}
200 OK
{
  "function": "remove_unix_deployment_credential",
  "response": {
    "unix_credentials": []

  }
```

add_windows_deployment_credential

Values: username, domain, password

Help: Adds a Windows deployment credential.

Example:

```
POST /fr-univ
{
        "function": "add_windows_deployment_credential",
        "values": {
                "username": "frestest",
                "domain": "testdomain",
                "password": "password123"
        }
}
200 OK
{
  "function": "add_windows_deployment_credential",
  "response": {
    "windows_credentials": [
      {
        "credentialid": "111fecc8-9609-44a6-8fe4-d376cdf1468c",
        "username": "frestest",
        "password": "********",
        "domain": "testdomain"
      }
    ]
  }
}
```

add_unix_deployment_credential

Values: username, password, rtpassword, key, assume

Help: Adds a unix deployment credential.

Example:

```
POST /fr-univ
{
        "function":"add_unix_deployment_credential",
        "values":{
                "username":"root or username",
                "password":"password",
                "rtpassword":"su password if assuming root using su",
                "key":"ssh key, optional, leave blank if not needed.",
                "assume": "su, sudo, or leave blank if not needed."
        }
}

200 OK
{
  "function": "add_unix_deployment_credential",
  "response": {
    "unix_credentials": [
      {
        "credentialid": "1a0464e1-e512-49c8-8b7f-edf32bf0e0b2",
        "username": "test",
        "password": "********",
        "rtpassword": "********",
        "key": "",
        "atype": 0,
        "utype": 0,
        "ptype": 0
      }
    ]
  }
}
```

deploy

Values: universal, subject, (optional: credentialid)

Help: Deploys and starts the subject software on a remote machine using a specific credential.

Example:

```
POST /fr-univ
{
        “function”:”deploy”,
        “values”:{
                “universal”:”192.168.0.1”,
                “subject”:”x64-win2k12-sub”,
**Optional** “credentialid": "1a0464e1-e512-49c8-8b7f-edf32bf0e0b2"

        }
}

200 OK
{
  "function": "deploy",
  "response": {
    "subject": "x64-win2k12-sub",
    "state": "running"
  }
}
```

undeploy

Values: universal, subject, (optional: credentialid)

Help: Undeploys the subject software from a remote machine using a specific credential.

Example:

```
POST /fr-univ
{
        “function”:”undeploy”,
        “values”:{
                “universal”:”192.168.0.1”,
                “subject”:”x64-win2k12-sub”,
**Optional** “credentialid": "1a0464e1-e512-49c8-8b7f-edf32bf0e0b2"

        }
}

200 OK
{
  "function": "undeploy",
  "response": {
    "subject": "x64-win2k12-sub",
    "state": "stopped"
  }
}
```

get_universal_subjects

Values: universal

Help: Returns a JSON array of Universal subjects.

Example:

```
POST /fr-univ
{
        “function”:”get_universal_subjects”,
        “values”:{
                “universal”:”192.168.0.1”
        }
}

200 OK
{
  "function": "get_universal_subjects",
  "response": {
    "subjects": [
      {
        "subjectid": "15369451274081149164",
        "name": "x64-win2k12-sub"
      }
    ],
    "universal": "192.168.0.1"
  }
}
```

export_gui_options

Values:

Help: Returns a JSON of the GUI Export options.

Example:

{

"function": "export_gui_options",

```
POST /fr-univ
{
        “function”:”export_gui_options”,
        “values”:{
        }
}

200 OK
{
  "response": {
    "universals": [
      "192.168.0.1"
    ],
    "platforms": [
      "Windows GUI",
      "Linux x64",
      "Linux x86",
      "OSX"
    ]
  }
}
```

export_msi_options

Values:

Help: Returns a JSON of the MSI Export options.

Example:

```
POST /fr-univ
{
       “function”:”export_msi_options”,
       “values”:{
       }
}

200 OK

{
  "function": "export_msi_options",
  "response": {
    "universals": [
      {
        "universal": "192.168.0.1",
        "adminport": 443,
        "port": 80,
        "username": "mshannon",
        "autostart_subject": 0,
        "servicename": "F-Response Universal",
        "servicedesc": "F-Response Universal Service",
        "exename": "f-response-univ-srv.exe",
        "waithint": 3600
      }
    ]
  }
}
```

get_deployment_credentials

Values:

Help: Returns a JSON of all deployment credentials.

Example:

```
POST /fr-univ
…
{
 “function”:”get_deployment_credentials”,
 “values”:{}
}

200 OK
{
  "function": "get_deployment_credentials",
  "response": {
    "unix_credentials": [
      {
        "credentialid": "1a0464e1-e512-49c8-8b7f-edf32bf0e0b2",
        "username": "test",
        "password": "********",
        "rtpassword": "********",
        "key": "",
        "atype": 0,
        "utype": 0,
        "ptype": 0
      }
    ],
    "windows_credentials": [
      {
        "credentialid": "111fecc8-9609-44a6-8fe4-d376cdf1468c",
        "username": "frestest",
        "password": "********",
        "domain": "testdomain"
      }
    ]
  }
}
```

list_universals

Values:

Help: Returns a JSON array of all Universal servers configured.

Example:

```
POST /fr-univ
…
{
        “function”:”list_universals”,
        “values”:{

        }
}

200 OK
{
  "function": "list_universals",
  "response": {
    "universals": [
      {
        "universal": "192.168.0.1",
        "adminport": 443,
        "port": 80,
        "username": "frestest",
        "connected": 0,
        "autoconnect": 0
      }    ]
  }
}
```

get_subject_targets

Values: universal, subjectid

Help: Returns a JSON array of F-Response Universal server subject targets.

Example:

```
POST /fr-univ
…
{
        “function”:”get_subject_targets”,
        “values”:{
            “universal”:”192.168.0.1”,
             “subjectid”:”15369451274081149164”

        }
}

200 OK
{
  "function": "get_subject_targets",
  "response": {
    "targets": [
      {
        "type": 3,
        "uuid": 8,
        "block_count": "10485760",
        "block_size": 512,
        "root_node": -842150451,
        "case_sensitive": 0,
        "name": "Disk-1-5120mb"
      },
      {
        "type": 3,
        "uuid": 7,
        "block_count": "83886080",
        "block_size": 512,
        "root_node": -842150451,
        "case_sensitive": 0,
        "name": "Disk-0-40960mb"
      },
      {
        "type": 3,
        "uuid": 6,
        "block_count": "2048",
        "block_size": 512,
        "root_node": -842150451,
        "case_sensitive": 0,
        "name": "Disk-0-Part-4-Unused-1mb"
      },
      {
        "type": 3,
        "uuid": 5,
        "block_count": "83165184",
        "block_size": 512,
        "root_node": -842150451,
        "case_sensitive": 0,
        "name": "Disk-0-Part-3-Active-40608mb"
      },
      {
        "type": 3,
        "uuid": 4,
        "block_count": "716800",
        "block_size": 512,
        "root_node": -842150451,
        "case_sensitive": 0,
        "name": "Disk-0-Part-2-Active-350mb"
      },
      {
        "type": 3,
        "uuid": 3,
        "block_count": "2048",
        "block_size": 512,
        "root_node": -842150451,
        "case_sensitive": 0,
        "name": "Disk-0-Part-1-Unused-1mb"
      },
      {
        "type": 3,
        "uuid": 2,
        "block_count": "83165184",
        "block_size": 512,
        "root_node": -842150451,
        "case_sensitive": 0,
        "name": "Volume-C-40608mb"
      },
      {
        "type": 2,
        "uuid": 1,
        "block_count": "10395647",
        "block_size": 4096,
        "root_node": 5,
        "case_sensitive": 0,
        "name": "DiscoveryShare-C"
      }
    ],
    "name": "x64-win2k12-sub",
    "universal": "192.168.0.1",
    "subjectid": "15369451274081149164",
    "available_drives": [
      "Z",
      "Y",
      "X",
      "L",
      "K",
      "J",
      "B"
    ]
  }
}
```

get_drive_letters

Values:

Help: Returns a JSON array of available drive letters.

Example:

```
POST /fr-univ
…
{
       “function”:”get_drive_letters”,
       “values”:{
       }
}

200 OK
{
  "function": "get_drive_letters",
  "response": {
    "available_drives": [
      "Z",
      "Y",
      "X",
      "K",
      "J",
      "I",
      "H",
      "B"
    ]
  }
}
```

set_universal_filter

Values:

Help: Enables filter mode and sets the Universal filter string.

Example:

```
POST /fr-univ
…
{
       “function”:”set_universal_filter”,
       “values”:{
             “universal”:”192.168.0.1”,
             “filter”:”x64”
       }
}

200 OK
{
  "function": "set_universal_filter",
  "response": {
    "filter": "x64"
  }
}
```