



TVDM documentation

Release

SECURity at the network EDge

March 31, 2016

1	Compute module	2
2	Config module	3
3	GraphInfo module	4
4	GraphInstatiator module	5
5	Network module	6
6	Orchestrator module	7
7	PSACConf module	8
8	PSAcreation module	9
9	diskManager module	10
10	libvirtManager module	11
11	manifestManager module	12
12	online_workflow_manager module	13
13	parserXML module	15
14	psar module	16
15	psarClient module	17
16	upr_client module	18
17	userTVD module	20
18	userTVDIPSECLess module	21
19	Indices and tables	22
	Python Module Index	23
	Index	24

Contents:

Compute module

```
class Compute.Compute (config)
```

```
    Bases: object
```

```
    deleteNF (nfID)
```

```
        Delete a NF nfID = ID to identify the NF on the NED
```

```
    instantiateNF (nfID, nfProperties)
```

```
        Instantiate a network function nfID = ID to identify the NF on the database  
        vmProperty = map with the properties of the VM: { memory:memory requirement(KiB)  
        vcpu: numeber of core needed interfaces:[list of interfaces] }
```

Config module

```
class Config.Configuration
    Bases: object

    ACCESS_INTERFACE
    DEBUG
    DNS_IP
    EXIT_INTERFACE
    GATEWAY_IP
    INSTANTIATED_VM_LOCATION
    LOG_FILE
    MANAGING_NETWORK_ADDRESS
    NED_ID
    NETMASK
    PSA_CONF_LOCATION
    PSA_MANI_LOCATION
    PSA_NETWORK_ADDRESS
    PSC_DP_IF_IP
    SCRIPT_LOCATION
    SPM_LOCATION
    TVDM_VERSION
    UPR_LOCATION
    USER_GRAPH_LOCATION
    USER_INTERFACE_NETWORK_ADDRESS
    USER_NETWORK_ADDRESS
    USE_LOCAL_FILE
    VERBOSE
    VM_IMAGES_LOCATION

    initialize()
```

GraphInfo module

Created on 04/set/2014

```
class GraphInfo.GraphInfo (instantiator, userGraphPath, config)  
    Bases: object  
    classdocs  
    get_client_address (environ)  
    on_get (request, response)
```

GraphInstatiator module

Created on 29/lug/2014

class GraphInstatiator.**GraphInstantiator** (*configure, logger, useIPSEC=False*)

Bases: object

Class used to instantiate the TVD and the Profile Graph

definePSC (*vlanID, mac*)

Define the template for the PSC of the TVD

deleteUser (*session*)

This function is used for the user log out. If there are multiple devices connected it will be deleted only the flows for the specific device that did the log out. If the last device for that specific user is disconnected the graph will be destroyed

instantiatePSA (*session*)

Instatiate the PSA of the TVD

instatiateTVD (*session*)

Intantiate the TVD for a logged user. In case for a TVD already instatiated it will generate the flows used for reedirecting the traffic of the new defice on it

signal_term_handler ()

Used on SIGTERM signal to destroy all the instantiated TVD

Network module

```
class Network.Network (configure)
    Bases: object

    configureNewIPOnDHCP (mac, token)
        Configure a new IP on the DHCP

    deleteFlow (bridgeName, flow)
        Delete a flow on the bridge

    deletePort (bridgeName, portName)
        Delete a port from a bridge

    generateFlow (bridgeName, flow)
        Generate a flow on the bridge

    generateMACaddress ()
        Generate a MAC address

    generatePort (bridgeName)
        Generate a port on a bridge

    generateRoutingTable (netInterface)
        Generate a new routing table

    generateVMPort ()
        Generate a port name for a VM

    getNewVLANID ()
        Return a new VLAN ID to use

    getPSAnewAddress ()
        Return a new IP address for a PSA
```

Orchestrator module

```
class Orchestrator.Orchestrator (instantiator)
    Bases: object

    Orchestrator class that intercept the REST call through the WSGI server

    on_delete (request, response)
    on_get (request, response)
    on_put (request, response)
```

PSAConf module

Created on 09/set/2014

class PSAConf.**PSAConf** (*confsPath, logger, config, instantiator*)

Bases: object

classdocs

get_client_address (*environ*)

on_get (*request, response, psa_id, conf_id*)

PSAcreation module

```
class PSAcreation.PSAcreation (instantiator)
    Bases: object

    Orchestrator class that intercept the REST call through the WSGI server

    get_client_address (environ)
    on_delete (request, response)
    on_get (request, response)
    on_put (request, response)
```

diskManager module

```
class diskManager.DiskManager (configure)
```

```
    Bases: object
```

```
    removeLocalDisk (localNFID)
```

```
        Remove a Disk from the local drive
```

```
    retriveDisk (nfID, localNFID)
```

```
        Retrive the Disk for a NF
```

libvirtManager module

```
class libvirtManager.ComputeManager (config)
    Bases: object

    defineXML (name, vmProperty)
        Define the xml to use to generate the VM name = name of the virtual machine vmProperty = map
        with the properties of the VM: { memory:memory requirement(KiB) vcpu:numerober of core needed
        disk:{location:disk location
            type:type of the disk}
        interfaces:[list of interfaces] }

    deleteNF (name)
        Destroy a VM

    getDiskInfo (nfID, localNFID)
        Get the information of the disk for the VM

    instantiateNF (name, vmProperty)
        Instantiate a VM
```

manifestManager module

```
class manifestManager.ManifestManager (config)  
    Bases: object  
    getManifest (psaID)  
        Get information of the manifest of the PSA
```

online_workflow_manager module

```
class online_workflow_manager.MyFormatter (fmt=None, datefmt=None)
    Bases: logging.Formatter

    Formats logs with particular time-stamps.

    converter ()
        timestamp[, tz] -> tz's local time from POSIX timestamp.

    formatTime (record, datefmt=None)

class online_workflow_manager.WorkflowManager (user, password, upr_url, spm_url, ned_ID,
                                                debug=False)
    Bases: object

    DEBUG = False

M2L (upr_url, spm_url, service_graph_xml, mspl_list_xml, user, password, editor_id)
    High-level function which translates the MSPL rules into low-level configurations. User is the username
    of the user. Password is the token of the user. Editor_id is the username of the user who created the policy.
    MSPL_list_xml is an XML list of MSPL rules which is returned by the SPM H2M service.

dbprint (string)
    Debug function. When enabled produces more verbose logs.

delete_all_rec_mspl (user, password, upr_url)
    Deletes old reconciled MSPLs which are no longer valid. User Profile Repository (UPR) must be specified
    as well as the access token.

get_PolicyStack (user, upr, cooperative, coop, uncoop)
    Get the policy stack of a particular user.

get_capability_from_mspl (mspl)
    Parse a specific MSPL and return a list of capabilities that are discovered inside

get_creator (user, upr)
    Get the administrator who created a given user

init_log ()
    Initialises logger and logs to a file called WFM.log

reconcile (user, password, upr_url, spm_url, ned_ID)
    Function which gets all the users associated to a particular user (e.g. stakedholders like parent, ISP, coun-
    try) and identify which policies are co-operative and non-co-operative. If successful, a reconciled applica-
    tion/service graph is produced and stored in the user's profile.
```


spm_reconcile (*spm_url, rec_input*)

Low-level auxiliary function which calls the SPM reconciliation service.

translate (*spm_url, mspl, psaID*)

Low-level auxiliary function which translates a single MSPL rule into a specific PSA configuration, where PSA ID is the name of the PSA.

`online_workflow_manager.main()`

Invoked when run manually from the command line

parserXML module

class `parserXML.parseXml`

Bases: `object`

static `parseAG` (*user_id*, *xml*)

Parse the application graph on a dictionary

psar module

```
class psar.Client (base_url)

    add_token_param (token, first_param)
    create_psa (name=None, token=None)
    delete_image (image_id, token=None)
    delete_manifest (image_id, token=None)
    delete_plugin (image_id, token=None)
    delete_psa (image_id, token=None)
    get_image_file (image_id, path, token=None)
    get_image_list (id=None, token=None)
    get_image_location (image_id, token=None)
    get_manifest (image_id, token=None)
    get_plugin_file (image_id, path, token=None)
    get_plugin_location (image_id, token=None)
    get_status (token=None)
    get_token (user, password, tenant, auth_URL)
    patch_image (image_id, new_status, token=None)
    put_image (image_id, name=None, token=None)
    put_image_file (image_id, path, disk_format, container_format, token=None)
    put_manifest_file (image_id, path, token=None)
    put_plugin (image_id, name=None, new_url=None, token=None)
    put_plugin_file (image_id, path, token=None)
    put_psarl_location (psarl_id, new_location, token=None)
```

psarClient module

```
class psarClient.Client (base_url)

    add_token_param (token, first_param)
    create_psa (name=None, token=None)
    delete_image (psa_id, token=None)
    delete_manifest (psa_id, token=None)
    delete_plugin (psa_id, token=None)
    delete_psa (psa_id, token=None)
    get_image_file (psa_id, path, token=None)
    get_image_list (id=None, token=None)
    get_image_location (psa_id, token=None)
    get_manifest (psa_id, path, token=None)
    get_plugin_file (psa_id, path, token=None)
    get_plugin_location (psa_id, token=None)
    get_status (token=None)
    get_token (user, password, tenant, auth_URL)
    patch_image (psa_id, new_status, token=None)
    put_image (psa_id, name=None, token=None)
    put_image_file (psa_id, path, disk_format, container_format, token=None)
    put_manifest_file (psa_id, path, token=None)
    put_plugin (psa_id, name=None, new_url=None, token=None)
    put_plugin_file (psa_id, path, token=None)
    put_psarl_location (psarl_id, new_location, token=None)
```

upr_client module

```

class upr_client.UPRClient (base_URL)

    associate_user_group (group, user_id)
    auth_user (user, password)
    create_group (name, description)
    create_mspl (target, editor, capability, is_reconciled, mspl, internalID=None)
    create_user (user_id, password, integrityLevel, type, is_cooperative, is_infrastructure, is_admin, creator=None, optimization_profile=None)
    delete_executed_psa (user_id, psa_id)
    delete_group (group_id)
    delete_group_psa (group, psa_id)
    delete_hspl (user_id, hspl_id)
    delete_mspl (mspl_id=None, target=None, editor=None, capability=None, is_reconciled=None)
    delete_reconciliation_report (user_id, ned_info=None)
    delete_user (user_id)
    delete_user_ag (target, editor)
    delete_user_group (user, group)
    delete_user_psa (user_id, psa_id)
    delete_user_psaconf (user_id, psa_id)
    delete_user_rag (user_id)
    get_created_users (user_id)
    get_executed_psa (user_id)
    get_group_psa (group)
    get_hspl (target=None, editor=None)
    get_mspl (internalID=None, target=None, editor=None, is_reconciled=False)
    get_reconciliation_report (user_id, ned_info=None)
    get_user_ag (target, editor=None)

```

```

get_user_creator (user_id)
get_user_groups (user_id)
get_user_list (user_id=None)
get_user_mspl_psa (user_id)
get_user_opt_profile (user_id)
get_user_psa (user_id, is_active=None)
get_user_psaconf (user_id, psa_id=None)
get_user_rag (user_id)
get_user_type (user_id)
list_group ()
list_user_group (group)
post_ag (target_id, editor_id, ag)
post_psaconf (user_id, psa_id, configuration)
post_rag (target_id, ned_info, rag)
post_reconciliation_report (user_id, ned_info, reconciliation_report)
put_executed_psa (user_id, psa_id)
put_group_psa (group, psa_id)
put_user_hspl (user_id, hspl, target)
put_user_mspl_psa (user_id, psa_id, mspl_id)
put_user_psa (user_id, data=None, psa_id=None, active=None, running_order=None)
    Data must be in the form accepted by the UPR { 'PSAList':[
        { 'psa_id':'12345', 'active':true, 'running_order':2
        }, {
            'psa_id':'54321', 'active':false, 'running_order':1
        }
    ]
    }
update_group (group, description=None)
update_user (user_id, name=None, password=None, integrityLevel=None, type=None,
    is_cooperative=None, is_infrastructure=None, is_admin=None, creator=None)

```

userTVD module

class userTVD.**UserTVD** (*userName, vlanID, networkManager, computeManager, config, logger*)

Bases: object

User TVD class in case of IPSEC NED configuration

addNewIP (*newIP*)

Add a new machine on the TVD with the given IP

associateIPPSA (*psaID*)

Associate an IP on a PSA

definePSA (*psaID*)

Define a new PSA for the TVD

delUserIP (*newIP*)

Remove the flow for the given IP

deleteAllTVD ()

Delete the TVD

deleteTVD (*IPaddr*)

Remove an IP from the TVD in case of the last IP the TVD will be deleted

generatePSCflows ()

Generate the flow for the PSC

instantiatePSA (*PSAList*)

Instantiate the PSAs for the TVD

setPSC (*newPSC, pscAddr*)

Configure the PSC of the TVD

userTVDIPSECLess module

class userTVDIPSECLess.**UserTVD** (*userName, vlanID, networkManager, computeManager, config, logger*)

Bases: object

User TVD class in case of IPSEC NED configuration

addNewIP (*newIP*)

Add a new machine on the TVD with the given IP

associateIPPSA (*psaID*)

Associate an IP on a PSA

definePSA (*psaID*)

Define a new PSA for the TVD

delUserIP (*newIP*)

Remove the flow for the given IP

deleteAllTVD ()

Delete the TVD

deleteTVD (*IPaddr*)

Remove an IP from the TVD in case of the last IP the TVD will be deleted

generatePSCflows ()

Generete the flow for the PSC

generateUserPSCflows (*userIP*)

Generete the flow for the PSC

instantiatePSA (*PSAList*)

Instantiate the PSAs for the TVD

setPSC (*newPSC, pscAddr*)

Configure the PSC of the TVD

To see how policies are downloaded, translated and reconciled, see:

Indices and tables

- `genindex`
- `modindex`
- `search`

c

Compute, [2](#)
Config, [3](#)

d

diskManager, [10](#)

g

GraphInfo, [4](#)
GraphInstatiator, [5](#)

l

libvirtManager, [11](#)

m

manifestManager, [12](#)

n

Network, [6](#)

o

online_workflow_manager, [13](#)
Orchestrator, [7](#)

p

parserXML, [15](#)
PSAConf, [8](#)
PSAcreation, [9](#)
psar, [16](#)
psarClient, [17](#)

u

upr_client, [18](#)
userTVD, [20](#)
userTVDIPSECLess, [21](#)

A

ACCESS_INTERFACE (Config.Configuration attribute), 3
 add_token_param() (psar.Client method), 16
 add_token_param() (psarClient.Client method), 17
 addNewIP() (userTVD.UserTVD method), 20
 addNewIP() (userTVDIPSECLess.UserTVD method), 21
 associate_user_group() (upr_client.UPRClient method), 18
 associateIPPSA() (userTVD.UserTVD method), 20
 associateIPPSA() (userTVDIPSECLess.UserTVD method), 21
 auth_user() (upr_client.UPRClient method), 18

C

Client (class in psar), 16
 Client (class in psarClient), 17
 Compute (class in Compute), 2
 Compute (module), 2
 ComputeManager (class in libvirtManager), 11
 Config (module), 3
 Configuration (class in Config), 3
 configureNewIPOnDHCP() (Network.Network method), 6
 converter() (online_workflow_manager.MyFormatter method), 13
 create_group() (upr_client.UPRClient method), 18
 create_mspl() (upr_client.UPRClient method), 18
 create_psa() (psar.Client method), 16
 create_psa() (psarClient.Client method), 17
 create_user() (upr_client.UPRClient method), 18

D

dbprint() (online_workflow_manager.WorkflowManager method), 13
 DEBUG (Config.Configuration attribute), 3
 DEBUG (online_workflow_manager.WorkflowManager attribute), 13
 definePSA() (userTVD.UserTVD method), 20
 definePSA() (userTVDIPSECLess.UserTVD method), 21

definePSC() (GraphInstantiator.GraphInstantiator method), 5
 defineXML() (libvirtManager.ComputeManager method), 11
 delete_all_rec_mspl() (online_workflow_manager.WorkflowManager method), 13
 delete_executed_psa() (upr_client.UPRClient method), 18
 delete_group() (upr_client.UPRClient method), 18
 delete_group_psa() (upr_client.UPRClient method), 18
 delete_hspl() (upr_client.UPRClient method), 18
 delete_image() (psar.Client method), 16
 delete_image() (psarClient.Client method), 17
 delete_manifest() (psar.Client method), 16
 delete_manifest() (psarClient.Client method), 17
 delete_mspl() (upr_client.UPRClient method), 18
 delete_plugin() (psar.Client method), 16
 delete_plugin() (psarClient.Client method), 17
 delete_psa() (psar.Client method), 16
 delete_psa() (psarClient.Client method), 17
 delete_reconciliation_report() (upr_client.UPRClient method), 18
 delete_user() (upr_client.UPRClient method), 18
 delete_user_ag() (upr_client.UPRClient method), 18
 delete_user_group() (upr_client.UPRClient method), 18
 delete_user_psa() (upr_client.UPRClient method), 18
 delete_user_psaconf() (upr_client.UPRClient method), 18
 delete_user_rag() (upr_client.UPRClient method), 18
 deleteAllTVD() (userTVD.UserTVD method), 20
 deleteAllTVD() (userTVDIPSECLess.UserTVD method), 21
 deleteFlow() (Network.Network method), 6
 deleteNF() (Compute.Compute method), 2
 deleteNF() (libvirtManager.ComputeManager method), 11
 deletePort() (Network.Network method), 6
 deleteTVD() (userTVD.UserTVD method), 20
 deleteTVD() (userTVDIPSECLess.UserTVD method), 21

- deleteUser() (GraphInstantiator.GraphInstantiator method), 5
- delUserIP() (userTVD.UserTVD method), 20
- delUserIP() (userTVDIPSECLess.UserTVD method), 21
- DiskManager (class in diskManager), 10
- diskManager (module), 10
- DNS_IP (Config.Configuration attribute), 3
- ## E
- EXIT_INTERFACE (Config.Configuration attribute), 3
- ## F
- formatTime() (online_workflow_manager.MyFormatter method), 13
- ## G
- GATEWAY_IP (Config.Configuration attribute), 3
- generateFlow() (Network.Network method), 6
- generateMACaddress() (Network.Network method), 6
- generatePort() (Network.Network method), 6
- generatePSCflows() (userTVD.UserTVD method), 20
- generatePSCflows() (userTVDIPSECLess.UserTVD method), 21
- generateRoutingTable() (Network.Network method), 6
- generateUserPSCflows() (userTVDIPSECLess.UserTVD method), 21
- generateVMPort() (Network.Network method), 6
- get_capability_from_mspl() (online_workflow_manager.WorkflowManager method), 13
- get_client_address() (GraphInfo.GraphInfo method), 4
- get_client_address() (PSAConf.PSAConf method), 8
- get_client_address() (PSAcreation.PSAcreation method), 9
- get_created_users() (upr_client.UPRClient method), 18
- get_creator() (online_workflow_manager.WorkflowManager method), 13
- get_executed_psa() (upr_client.UPRClient method), 18
- get_group_psa() (upr_client.UPRClient method), 18
- get_hspl() (upr_client.UPRClient method), 18
- get_image_file() (psar.Client method), 16
- get_image_file() (psarClient.Client method), 17
- get_image_list() (psar.Client method), 16
- get_image_list() (psarClient.Client method), 17
- get_image_location() (psar.Client method), 16
- get_image_location() (psarClient.Client method), 17
- get_manifest() (psar.Client method), 16
- get_manifest() (psarClient.Client method), 17
- get_mspl() (upr_client.UPRClient method), 18
- get_plugin_file() (psar.Client method), 16
- get_plugin_file() (psarClient.Client method), 17
- get_plugin_location() (psar.Client method), 16
- get_plugin_location() (psarClient.Client method), 17
- get_PolicyStack() (online_workflow_manager.WorkflowManager method), 13
- get_reconciliation_report() (upr_client.UPRClient method), 18
- get_status() (psar.Client method), 16
- get_status() (psarClient.Client method), 17
- get_token() (psar.Client method), 16
- get_token() (psarClient.Client method), 17
- get_user_ag() (upr_client.UPRClient method), 18
- get_user_creator() (upr_client.UPRClient method), 18
- get_user_groups() (upr_client.UPRClient method), 19
- get_user_list() (upr_client.UPRClient method), 19
- get_user_mspl_psa() (upr_client.UPRClient method), 19
- get_user_opt_profile() (upr_client.UPRClient method), 19
- get_user_psa() (upr_client.UPRClient method), 19
- get_user_psaconf() (upr_client.UPRClient method), 19
- get_user_rag() (upr_client.UPRClient method), 19
- get_user_type() (upr_client.UPRClient method), 19
- getDiskInfo() (libvirtManager.ComputeManager method), 11
- getManifest() (manifestManager.ManifestManager method), 12
- getNewVLANID() (Network.Network method), 6
- getPSAnewAddress() (Network.Network method), 6
- GraphInfo (class in GraphInfo), 4
- GraphInfo (module), 4
- GraphInstantiator (class in GraphInstantiator), 5
- GraphInstantiator (module), 5
- ## I
- init_log() (online_workflow_manager.WorkflowManager method), 13
- initalize() (Config.Configuration method), 3
- INSTANTIATED_VM_LOCATION (Config.Configuration attribute), 3
- instantiateNF() (Compute.Compute method), 2
- instantiateNF() (libvirtManager.ComputeManager method), 11
- instantiatePSA() (GraphInstantiator.GraphInstantiator method), 5
- instantiatePSA() (userTVD.UserTVD method), 20
- instantiatePSA() (userTVDIPSECLess.UserTVD method), 21
- instatiateTVD() (GraphInstantiator.GraphInstantiator method), 5
- ## L
- libvirtManager (module), 11
- list_group() (upr_client.UPRClient method), 19
- list_user_group() (upr_client.UPRClient method), 19
- LOG_FILE (Config.Configuration attribute), 3

M

M2L() (online_workflow_manager.WorkflowManager method), 13
 main() (in module online_workflow_manager), 14
 MANAGING_NETWORK_ADDRESS (Config.Configuration attribute), 3
 ManifestManager (class in manifestManager), 12
 manifestManager (module), 12
 MyFormatter (class in online_workflow_manager), 13

N

NED_ID (Config.Configuration attribute), 3
 NETMASK (Config.Configuration attribute), 3
 Network (class in Network), 6
 Network (module), 6

O

on_delete() (Orchestrator.Orchestrator method), 7
 on_delete() (PSAcreation.PSAcreation method), 4
 on_get() (GraphInfo.GraphInfo method), 4
 on_get() (Orchestrator.Orchestrator method), 7
 on_get() (PSAConf.PSAConf method), 8
 on_get() (PSAcreation.PSAcreation method), 9
 on_put() (Orchestrator.Orchestrator method), 7
 on_put() (PSAcreation.PSAcreation method), 9
 online_workflow_manager (module), 13
 Orchestrator (class in Orchestrator), 7
 Orchestrator (module), 7

P

parseAG() (parserXML.parseXml static method), 15
 parserXML (module), 15
 parseXml (class in parserXML), 15
 patch_image() (psar.Client method), 16
 patch_image() (psarClient.Client method), 17
 post_ag() (upr_client.UPRClient method), 19
 post_psaconf() (upr_client.UPRClient method), 19
 post_rag() (upr_client.UPRClient method), 19
 post_reconciliation_report() (upr_client.UPRClient method), 19
 PSA_CONF_LOCATION (Config.Configuration attribute), 3
 PSA_MANI_LOCATION (Config.Configuration attribute), 3
 PSA_NETWORK_ADDRESS (Config.Configuration attribute), 3
 PSAConf (class in PSAConf), 8
 PSAConf (module), 8
 PSAcreation (class in PSAcreation), 9
 PSAcreation (module), 9
 psar (module), 16
 psarClient (module), 17
 PSC_DP_IF_IP (Config.Configuration attribute), 3

put_executed_psa() (upr_client.UPRClient method), 19
 put_group_psa() (upr_client.UPRClient method), 19
 put_image() (psar.Client method), 16
 put_image() (psarClient.Client method), 17
 put_image_file() (psar.Client method), 16
 put_image_file() (psarClient.Client method), 17
 put_manifest_file() (psar.Client method), 16
 put_manifest_file() (psarClient.Client method), 17
 put_plugin() (psar.Client method), 16
 put_plugin() (psarClient.Client method), 17
 put_plugin_file() (psar.Client method), 16
 put_plugin_file() (psarClient.Client method), 17
 put_psarl_location() (psar.Client method), 16
 put_psarl_location() (psarClient.Client method), 17
 put_user_hspl() (upr_client.UPRClient method), 19
 put_user_mspl_psa() (upr_client.UPRClient method), 19
 put_user_psa() (upr_client.UPRClient method), 19

R

reconcile() (online_workflow_manager.WorkflowManager method), 13
 removeLocalDisk() (diskManager.DiskManager method), 10
 retrieveDisk() (diskManager.DiskManager method), 10

S

SCRIPT_LOCATION (Config.Configuration attribute), 3
 setPSC() (userTVD.UserTVD method), 20
 setPSC() (userTVDIPSECLess.UserTVD method), 21
 signal_term_handler() (GraphInstantiator.GraphInstantiator method), 5
 SPM_LOCATION (Config.Configuration attribute), 3
 spm_reconcile() (online_workflow_manager.WorkflowManager method), 13

T

translate() (online_workflow_manager.WorkflowManager method), 14
 TVDM_VERSION (Config.Configuration attribute), 3

U

update_group() (upr_client.UPRClient method), 19
 update_user() (upr_client.UPRClient method), 19
 upr_client (module), 18
 UPR_LOCATION (Config.Configuration attribute), 3
 UPRClient (class in upr_client), 18
 USE_LOCAL_FILE (Config.Configuration attribute), 3
 USER_GRAPH_LOCATION (Config.Configuration attribute), 3
 USER_INTERFACE_NETWORK_ADDRESS (Config.Configuration attribute), 3
 USER_NETWORK_ADDRESS (Config.Configuration attribute), 3

UserTVD (class in userTVD), [20](#)

UserTVD (class in userTVDIPSECLess), [21](#)

userTVD (module), [20](#)

userTVDIPSECLess (module), [21](#)

V

VERBOSE (Config.Configuration attribute), [3](#)

VM_IMAGES_LOCATION (Config.Configuration attribute), [3](#)

W

WorkflowManager (class in online_workflow_manager),
[13](#)