common/tools
converters
NumeralConverter
network
BinaryTools
common/type_utils
binary
Octet
binary_factory
OctetFlyWeightFactory
common/design_template/cor
BiDirectionalCallHandler
SimpleCallHandler
__init__.py
<interface>>
MACType
UNICAST = 1
MULTICAST = 2
BROADCAST = 3
mac_classifiers
<interface>>
MACAddrClassifierHandler
next_handler: SimpleCallHandler = None
+ <<override>> verify_type(request:List[Octet]): bool
+ <<override>> handle(request:Any): List[Octet] | MACAddrClassifierHandler
MACAddrClassifier
+ <<static>> classify_mac(mac: List[BinaryClass], classifiers: List[MACAddrClassifierHandler]=None): MACType
BroadcastMACAddrClassifierHandler
+ <<override>> verify_type(request:List[Octet]): bool
+ <<override>> handle(request:Any): List[Octet] | MACAddrClassifierHandler
UnicastMACAddrClassifierHandler
+ <<override>> verify_type(request:List[Octet]): bool
+ <<override>> handle(request:Any): List[Octet] | MACAddrClassifierHandler
MulticastMACAddrClassifierHandler
+ <<override>> verify_type(request:List[Octet]): bool
+ <<override>> handle(request:Any): List[Octet] | MACAddrClassifierHandler
mac_converter
<interface>>
MACConverterHandler
next_handler: SimpleCallHandler = None
+ set_padding(padding: bool): void
+ <<abstract>> to_octets(request: Any): List[Octet]
+ <<abstract>> handle(request:Any): List[Octet] | MACConverterHandler
MACConverter
+ convert_mac(mac: Any, converters: List[MACConverterHandler]=None): List[Octet]
+ convert_oui(mac: Any, converters: List[MACConverterHandler]=None): List[Octet]
+ convert_to_octet(mac: Any): List[Octet]
BinaryClassMACAddrConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
BinaryClassOUI24ConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
BinaryDigitMACAddrConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
BinaryDigitOUI24ConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
DashedHexMACAddrConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
DashedHexOUI24ConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
ColonHexMACAddrConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
ColonHexOUI24ConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
DotHexMACAddrConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
DotHexOUI24ConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
BinaryCiscoUI24ConverterHandler
+ <<override>> to_octets(request:Any): List[Octet]
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
mac_address
<interface>>
InterfaceMACAddr
- address: List[Octet] = []
- oui: Union[Octet, None] = None
- mac_type: MACType = None
- oui_database = LocalOUIDatabase()
+ __init__(): void
- <<abstract>> initialize()
- <<abstract>> validate(mac: List[Octet]): List[Octet]
+ <<abstract>> classify_mac_address(): void
+ <<abstract>> search_oui(): void
+ <<static>> <<property>> binary_digit(): List[int]
+ <<static>> <<property>> binary_string(): str
+ __str__(): str
MACAddr
...
mac_utils
<Enum>>
OUIType
UNKNOWN = 0
IAB = 1
MA_S = 2
MA_M = 3
MA_L = 4
CID = 5
OUIUtils
- size: Dicts = {}
- conditions = List[BinaryClass] {}
- mask: List[BinaryClass] {}
- OUIType: OUIType
- organization: str | None
- mac_range: str | None
- OUIHex: str | None
- address: str | None
- _instance = None
+ jfy.exe(cls): object
+ __new__(): void
+ {property}record(): dict
+ validate()
OUIUnitCreator
+ create_product(**kwargs): OUIUnit
<Enum>>
OUIORStrategy
- SWIPE_ITEM/TOON = 0
TRIE = 1
<Enum>>
OUIMask
- IAB = <<int(0xfffff)>>
- MA_S = <<int(0xffffff)>>
- MA_M = <<int(0xfffff0)>>
- MA_L = <<int(0xffffff)>>
- CID = <<int(0xfffff)>>
oui_db/database
<interface>>
OUIDatabase
+ <<abstract>> __init__(**kwargs): void
+ <<abstract>> load(): void
+ <<abstract>> update(**args): void
+ <<abstract>> reset(*args): void
+ <<abstract>> search(mac: Any): OUIUnit | None
+ <<abstract>> bulk_search(mac: List[Any]) -> Dict[str, OUIUnit]
OUIDatabase <<Singleton>>
+ <<override>> __init__(**kwargs): void
+ <<override>> load(): void
+ <<override>> update(file_path: str): void
+ <<override>> reset(updater_type: OUIType): void
+ <<override>> search(mac: Any): OUIUnit | None
+ <<override>> bulk_search(mac: List[Any]) -> Dict[str, OUIUnit]
oui_db/searcher
<interface>>
SearcherStrategy
+ <<abstract>> search(*args): void
<interface>>
OUIDBSearcher
+ <<abstract>> set_strategy(strategy: OUIDBStrategy): SearcherStrategy
+ <<abstract>> search(mac: List[Octet], database: list): OUIUnit
LocalOUIDBSearcher
- searcher_type: OUIType = OUIType.Unknown
- mask: List[Octet] = []
+ __init__(strategy: OUIDBStrategy = OUIDBStrategy.TRIE)
+ <<override>> set_strategy(strategy: OUIDBStrategy): SearcherStrategy
+ <<override>> search(mac: List[Octet], database: list): OUIUnit
+ <<static>> <<_writeOnce>> List[Octet]: oui: OUIUnit(): bool
LocalIabSearcher
+ <<override>> searcher_type = OUIType.IAB
+ <<override>> mask: List[Octet] = OUIMask.IAB.value
LocalMaSearcher
+ <<override>> searcher_type = OUIType.MA_S
+ <<override>> mask: List[Octet] = OUIMask.MA_S.value
LocalMamSearcher
+ <<override>> searcher_type = OUIType.MA_M
+ <<override>> mask: List[Octet] = OUIMask.MA_M.value
LocalMalSearcher
+ <<override>> searcher_type = OUIType.MA_L
+ <<override>> mask: List[Octet] = OUIMask.MA_L.value
LocalCidSearcher
+ <<override>> searcher_type = OUIType.CID
+ <<override>> mask: List[Octet] = OUIMask.CID.value
SimpleSearcherStrategy
+ is_with(mac: List[Octet], oui: OUIUnit): bool
+ <<override>> search(mac:List[Octet], oui_data:list): OUIUnit
TrieSearcherStrategy
+ <<override>> search(mac:List[Octet], oui_data:list): OUIUnit
oui_db/updater
<interface>>
OUIDBUpdater
+ <<abstract>> update(new_official_doc: str): void
+ <<abstract>> reset(): void
LocalOUIDBUpdater
- base_dir: str = <default value>
- custom_db_dir = None
- updater_type: OUIType = OUIType.UNKNOWN
- file_parsers: List = []
+ <<property>> updater_type(): OUIType
+ <<override>> update(new_official_doc: str): void
+ <<override>> reset(): void
LocalIabUpdater
+ <<override>> custom_db_dir = custom_iab.json
+ <<override>> updater_type: OUIType = OUIType.IAB
+ <<override>> file_parsers = [
IabOuiTxtFileParserHandler,
IabOuiCsvFileParserHandler,
]
LocalMaSUpdater
+ <<override>> custom_db_dir = custom_mas.json
+ <<override>> updater_type: OUIType = OUIType.MA_S
+ <<override>> file_parsers = [
MasOuiTxtFileParserHandler,
MasOuiCsvFileParserHandler
]
LocalMamUpdater
+ <<override>> custom_db_dir = custom_mam.json
+ <<override>> updater_type: OUIType = OUIType.MA_M
+ <<override>> file_parsers = [
MamOuiTxtFileParserHandler,
MamOuiCsvFileParserHandler
]
LocalMalUpdater
+ <<override>> custom_db_dir = custom_mal.json
+ <<override>> updater_type: OUIType = OUIType.MA_L
+ <<override>> file_parsers = [
MalOuiTxtFileParserHandler,
MalOuiCsvFileParserHandler
]
LocalCidUpdater
+ <<override>> custom_db_dir = custom_cid.json
+ <<override>> updater_type: OUIType = OUIType.CID
+ <<override>> file_parsers = [
CidOuiTxtFileParserHandler,
CidOuiCsvFileParserHandler,
]
oui_db/loaders
<interface>>
OUIDBLoader
+ <<abstract>> set_strategy(strategy: OUIDBStrategy): LoaderStrategy
+ <<abstract>> connect()
+ <<abstract>> load(): Dict[str, Dict]
LocalOUIDBLoader
- base_dir: str = <default value>
- default_db: str = None
- custom_db: str = None
- loader_type: OUIType = OUIType.UNKNOWN
- official_docs: List[str] = []
- file_parsers: List = []
- data: Dict = {}
+ __init__(strategy:OUIDBStrategy = OUIDBStrategy.TRIE): void
+ <<override>> set_strategy(strategy: OUIDBStrategy): LoaderStrategy
- initialize(): void
+ <<property>> data(): Dict['OUI': str, 'oui_units': List[OUIUnit]]
+ <<property>> loader_type(): OUIType
+ <<override>> connect(*args): bool
+ <<override>> load(*args): void
- compare_hash(): bool
- create_default_db(): void
TrieOUIUnit
+ <<override>> __init__():void
LoaderStrategy
+ <<abstract>> load(*args): List[OUIUnit]
SimpleLoaderStrategy
+ <<override>> load(oui_units): List[Dict]: List[OUIUnit]
TrieLoaderStrategy
+ __init__(): void
- insert(oui_id: str, oui_unit: dict): void
+ <<override>> load(oui_units: List[Dict]): List[OUIUnit]
oui_file_parsers
<interface>>
OUITxtFileParserHandler
file: IEEEOuiFiles | None = None
- mask: List[Octet] = []
- OUIType: OUIType = OUIType.UNKNOWN
+ <<override>> handle(request:Any): List[Octet] | MACConverterHandler
+ <<abstract>> parse(oui_doc: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]]
+ <<abstract>> parse_line(_range(line: List[Octet]): mac_mask: List[Octet]): List[str]
+ <<abstract>> parse_physical_address(*args: str): str
+ <<property>> file(): IEEEOuiFiles
- generate_file_info(remote/oui_doc_path:str):void
IEEEOuiFiles
+ <<override>> validate(): void
+ <<override>> read(): void
<interface>>
OUICsvFileParserHandler
+ <<override>> parse_mac_range(mac: List[Octet], mac_mask: List[Octet]): List[str]
+ <<override>> parse_physical_address(address_line1: str, address_line2: str, country: str): str
IabOuiTxtFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> oui_type: OUIType: IAB
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
IabOuiCsvFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> OUIType: OUIType: IAB
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
MasOuiTxtFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> oui_type: OUIType: MA_S
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
MasOuiCsvFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> OUIType: OUIType: MA_S
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
MamOuiTxtFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> oui_type: OUIType: MA_M
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
MamOuiCsvFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> OUIType: OUIType: MA_M
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
MalOuiTxtFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> oui_type: OUIType: MA_L
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
MalOuiCsvFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> OUIType: OUIType: MA_L
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
CidOuiTxtFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> oui_type: OUIType: CID
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
CidOuiCsvFileParserHandler
+ <<override>> mask: List[Octet]
+ <<override>> OUIType: OUIType: CID
+ <<override>> parse(oui_doc: str): Dict[str, List[OUIUnit]]
+ <<override>> handle(oui_doc_path: str): dict['oui': str, 'type': OUIType, 'oui_units': List[OUIUnit]] | OUIFileParserHandler
File/file_utils
File