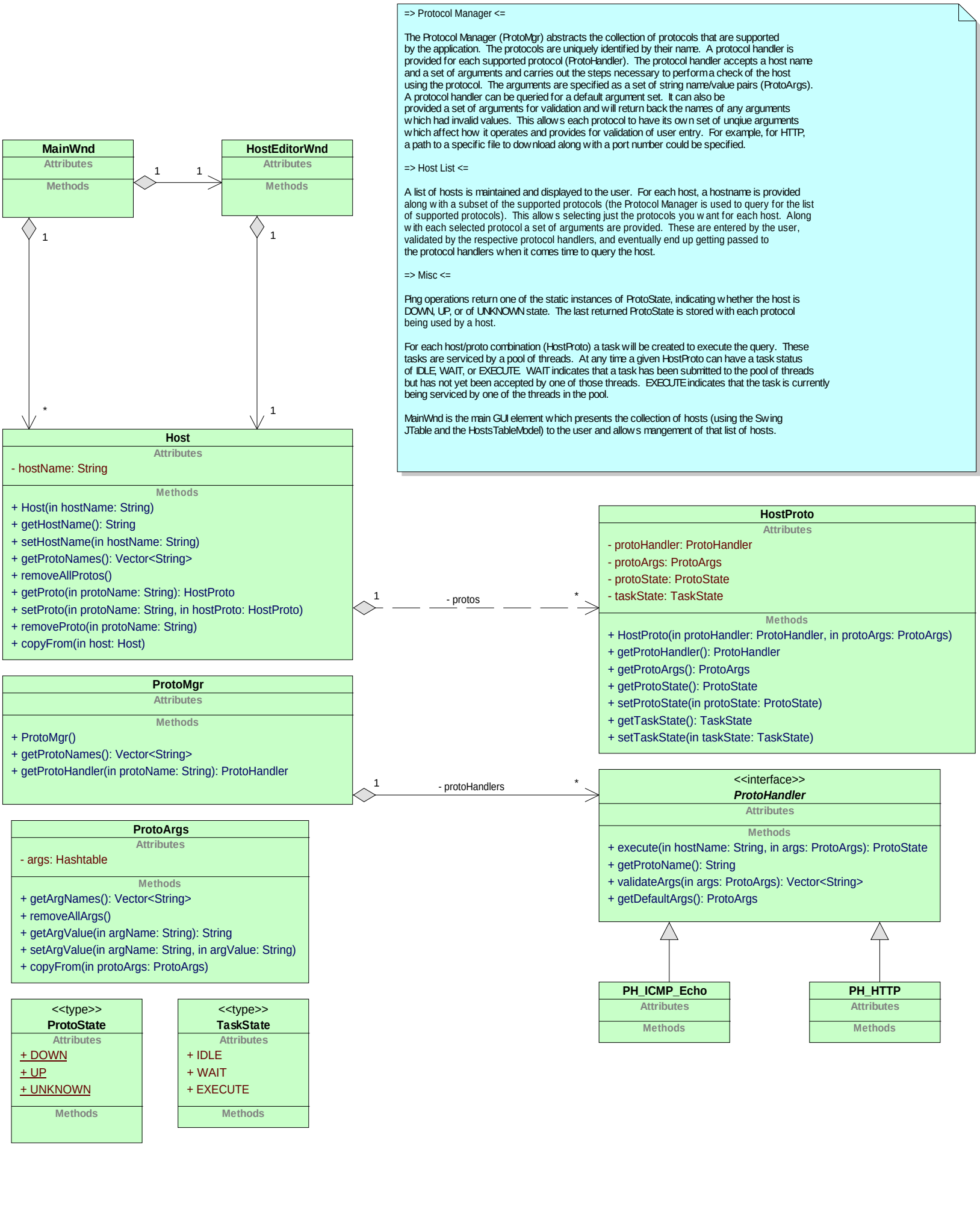
=> Protocol Manager <=
The Protocol Manager (ProtoMgr) abstracts the collection of protocols that are supported by the application. The protocols are uniquely identified by their name. A protocol handler is provided for each supported protocol (ProtoHandler). The protocol handler accepts a host name and a set of arguments and carries out the steps necessary to perform a check of the host using the protocol. The arguments are specified as a set of string name/value pairs (ProtoArgs). A protocol handler can be queried for a default argument set. It can also be provided a set of arguments for validation and will return back the names of any arguments which had invalid values. This allows each protocol to have its own set of unqiue arguments which affect how it operates and provides for validation of user entry. For example, for HTTP, a path to a specific file to download along with a port number could be specified.
=> Host List <=
A list of hosts is maintained and displayed to the user. For each host, a hostname is provided along with a subset of the supported protocols (the Protocol Manager is used to query for the list of supported protocols). This allows selecting just the protocols you want for each host. Along with each selected protocol a set of arguments are provided. These are entered by the user, validated by the respective protocol handlers, and eventually end up getting passed to the protocol handlers when it comes time to query the host.
=> Misc <=
Ping operations return one of the static instances of ProtoState, indicating whether the host is DOWN, UP, or of UNKNOWN state. The last returned ProtoState is stored with each protocol being used by a host.
For each host/proto combination (HostProto) a task will be created to execute the query. These tasks are serviced by a pool of threads. At any time a given HostProto can have a task status of IDLE, WAIT, or EXECUTE. WAIT indicates that a task has been submitted to the pool of threads but has not yet been accepted by one of those threads. EXECUTE indicates that the task is currently being serviced by one of the threads in the pool.
MainWnd is the main GUI element which presents the collection of hosts (using the Swing JTable and the HostsTableModel) to the user and allows mangement of that list of hosts.
MainWnd
Attributes
Methods
HostEditorWnd
Attributes
Methods
1
1
1
1
*
1
Host
Attributes
- hostName: String
Methods
+ Host(in hostName: String)
+ getHostName(): String
+ setHostName(in hostName: String)
+ getProtoNames(): Vector<String>
+ removeAllProtos()
+ getProto(in protoName: String): HostProto
+ setProto(in protoName: String, in hostProto: HostProto)
+ removeProto(in protoName: String)
+ copyFrom(in host: Host)
1
- protos
*
HostProto
Attributes
- protoHandler: ProtoHandler
- protoArgs: ProtoArgs
- protoState: ProtoState
- taskState: TaskState
Methods
+ HostProto(in protoHandler: ProtoHandler, in protoArgs: ProtoArgs)
+ getProtoHandler(): ProtoHandler
+ getProtoArgs(): ProtoArgs
+ getProtoState(): ProtoState
+ setProtoState(in protoState: ProtoState)
+ getTaskState(): TaskState
+ setTaskState(in taskState: TaskState)
ProtoMgr
Attributes
Methods
+ ProtoMgr()
+ getProtoNames(): Vector<String>
+ getProtoHandler(in protoName: String): ProtoHandler
1
- protoHandlers
*
<<interface>>
ProtoHandler
Attributes
Methods
+ execute(in hostName: String, in args: ProtoArgs): ProtoState
+ getProtoName(): String
+ validateArgs(in args: ProtoArgs): Vector<String>
+ getDefaultArgs(): ProtoArgs
ProtoArgs
Attributes
- args: Hashtable
Methods
+ getArgNames(): Vector<String>
+ removeAllArgs()
+ getArgValue(in argName: String): String
+ setArgValue(in argName: String, in argValue: String)
+ copyFrom(in protoArgs: ProtoArgs)
PH_ICMP_Echo
Attributes
Methods
PH_HTTP
Attributes
Methods
<<type>>
ProtoState
Attributes
+ DOWN
+ UP
+ UNKNOWN
Methods
<<type>>
TaskState
Attributes
+ IDLE
+ WAIT
+ EXECUTE
Methods