Device Management

Commands

This page references all generic commands used by the platform. The commands are serialized and sent using M3DA Protocol.

This is an "applicative level" specification opposed to the "serialization level" specification from the M3DA Protocol document.

In order to send command from the server to the device, or from the device to the server, the **M3DA::Command** obj ect is used and encapsulated into an **M3DA::Message** object as stated into the protocol specification.

Command Name	Command Arguments	Comments
Main		
ReadNode	path.of.the.node.to.read (string)	Read a node and its children (if any) from a tree. The node can be either terminal (a leaf: to get the value of a property) or have sub nodes. In the later case the content of the node and all its sub nodes will be transmitted. This command provokes the sending of Data Messages that will contain the data associated to that node. The data of the message is composed of a map that hold the properties as key, and their value as value. If a node contains a child, Data Messages will be sent recursively in order to present all the data of the sub nodes as well. If the specified node does not exist or contains no data, no message is sent. **Note**: *path.of.the.node.to.read* is a path that is not necessarily linked to the device or asset topology. In particular, the assetID is not repeated in that path. Ex: Request bearer configuration it should use: "ReadNode config.network.bearers"

Connect		Ask the M2M Agent to connect to the platform server. The command must be addressed to the device (M3DA::Message.path = "@sys").
Reboot		Ask the device or one of its asset to reboot. The device or the asset is designated thanks to the path of the M3DA::Message that wrap this command.

ResetToFactoryDefault	-> restart (boolean|number)	Reset agent settings to factory defaults. All persisted data about agent settings and installed software are lost.
		
		Impacted modules/functionalities:
		• ReadyAgent persisted **config** is reset to **defaultconfig**
		(depending on the differences between defaultconfig.lua and persisted config, this operation may impact: server url, hearbeat, ..., or any ReadyAgent config parameter)
		• Treemgr mapping are reset: it will be regenerated from .map files on next boot
		• **Persisted AWTDA data**: asset and device data are erased
		• **Applications** installed in ApplicationContainer are erased
		• Update module:
		  • deletion of any update in progress
		  • **software version list** is cleared
		
		Not impacted modules/functionalities:
		• AWTDA3 security credentials are **not** cleared

		Note: • This command is dependent on ReadyAgent integration, so ResetToFactoryDefault command implementation can differ from a device to another to fit integration needs, ReadyAgent product provides a generic implementation matching previous remarks -> If this is a boolean value and it is true, it requests the agent to be restarted with a default timeout (6 seconds) -> If this is a number and it is greater than zero, it requests the agent to be restarted in "restart" seconds
Software Update		
ExecuteScript	-> url (string) -> signature (string)	-> url to retrieve the Lua script The Lua script can be either Lua source file or precompiled Lua bytecode file. -> signature of Lua Script The signature will fit the security level defined within the ReadyAgent. First step: Signature will be **MD5 hash**, and will be sent in **hexa** in an ascii string.

SoftwareUpdate	-> url (string) -> signature (string)	-> url provided by the server where the Software Update Package can be downloaded. Must **not** end by a trailing "/" character, unless archive name contains one (not recommended) -> signature of the Software Update Package The signature will fit the security level defined within the ReadyAgent. First step: Signature will be **MD5 hash**, and will be sent in **hexa** in an ascii string which size must be 32 chars (prefixing zeros chars must be sent!), and in lower case only.
TCP Remote Connection		
TCPRemoteConnect		Install a TCP tunnel
Log Upload		
LogUpload	-> url (string) -> log type (string)	The url where the logs are to be uploaded. Has to be of the form "ftp://" to request ftp upload, else "http://" for HTTP Post upload string equal to: "ram" to retrieve logs in ram (i.e. only from current ReadyAgent execution), or "flash" to get the logs from flash space Note: The content of flash or ram buffer depends on the log policy defined in ReadyAgent Config
Application Container		
appcon.start	-> appname (string)	Start an application
appcon.stop	-> appname (string)	Stop an application
appcon.autostart	-> appname (string) -> autostart (boolean)	Configure an application to start automatically or not.

Variables

Variable	Read/Write	Description

@sys.appcon.list	RO	list of all applications currently managed by appcon, as a single string of space-separated names
@sys.appcon.apps.<appname>.started	RW	whether the application is currently started (Boolean)
@sys.appcon.apps.<appname>.autostart	RW	whether the application starts automatically (Boolean)
@sys.appcon.apps.<appname>.runnable	RO	whether it is a runnable application
@sys.appcon.apps.<appname>.<daemonattr>	RO	The current value of every daemon attribute <daemonattr>. Current attributes include: appname, priviledged, prog, wd, pid, startcount, lastexittype, lastexitcode

TBCompleted

Events

TBD