

The CMDLi Interpreter

The cmdli.rc file

When the CMDLi interpreter initializes it looks for an optional runtime configuration file named “cmdli.rc” in its current directory, and loads it if it exists. The file syntax is a standard key=value syntax, with one entry per line. Double quotes are necessary around values that are not a simple number. For example, 42.7 is fine without quoting, but “234.5.6.7:5000” requires quotes. Normal C /*comment*/ and C++ //comment styles are allowed, as well as using the sharp character (#) in the first column to make the entire line a comment. The keys ARE case sensitive, and whitespace is ignored, but the entire entry must be on a single line. Currently recognized entries are:

MulticastAddr

Tells the interpreter to use the specified network address for multicasting, instead of the default address of 224.0.1.150:12345 to talk to the other CMDLi interpreters. Note that the default address is a valid multicasting address for the JBoxes and this entry is only required in other circumstances.

Syntax:

```
MulticastAddr = “IPaddress:Port”
```

Example:

```
MulticastAddr = “234.5.6.7:5000”
```

RobotAddresses

The default behavior is that the CMDLi interpreters communicate using multicasting. However, this entry allows converting that behavior into replicated unicast messages. That is, instead of periodically sending a single multicast UDP packet to all other interpreters, each interpreter will instead send N-1 copies of the packet, each directed to one of the addresses listed in this entry. This is normally used if multicasting is not working, due to network topology, or to allow running multiple instances of the interpreter on a single computer for testing or demos. Note that all addresses should be listed, including the one for the local interpreter. It will remove its own address from the list before sending messages. Any number of addresses can be listed, but they have to all be on the same line in the file.

Syntax:

```
RobotAddresses = “IPaddress:Port”, “IPaddress:Port”, “IPaddress:Port”
```

Example:

```
RobotAddresses = "192.168.1.1:5000", "192.168.1.2:5000", 192.168.1.2:5001"
```

UnicastPort

Specifies the port the interpreter should listen on for unicast messages. The default port number is 12345, although unicast messages are not used by default and this command is normally used in conjunction with a **RobotAddresses** entry to specify the port for this instance. Legal port numbers are between 1 and 65535 inclusive, with reasonable values between 4000 and 65000

Syntax:

```
UnicastPort = "Portnumber"
```

Example:

```
UnicastPort = 5000
```

CommsDebug

Used to enable debugging messages from the communications library. The value is a bit mask to allow selectively enabling different classes of messages. To create a value, add the individual bits together to form an overall debug value. The available message classes are:

- 1: General - provides setup details and runtime warnings
- 2: Transmitter - provides details about each message sent
- 4: Receiver - details about each message received
- 8: Detailed Receiver – provides details about each received packet
- 16: Raw – prints the raw data contained in each received and transmitted packet

Syntax:

```
CommsDebug = value
```

Example:

```
// Turn on General, Receiver, and Raw messages (1 + 4 + 16)  
CommsDebug = 21
```

Debug

Used to enable debugging messages from the CMDL interpreter. The value is a bit mask to allow selectively enabling different classes of messages. To create a value, add the individual bits together to form an overall debug value. The available message classes are:

- 1: Loader General - Extra details and warnings while loading the CMDL mission
- 2: Loader Parser – Detailed parsing messages while loading the CMDL mission
- 4: Loader Scanner – Detailed scanning messages while loading the CMDL mission

- 8: Interpreter – Provides detailed status while executing the CMDL mission
- 16: Synchronization – Details of the inter-interpreter synchronization process

Syntax:

```
Debug = value
```

Example:

```
// Turn on General, Interpreter, and Synchronization messages (1 + 8 + 16)  
Debug = 25
```

CommsTimeout

When multiple robots must synchronize at a point in the mission, it can happen that one of them fails, for example due to losing radio communications or a power failure. After the `commsTimeout` period, the remainder of the robots will declare the failed robot dead, and begin ignoring it. If they later again receive status messages from the failed robot, they will pause at the end of their current mission segment and wait for it to catch up. This entry sets the length of the default `commsTimeout` period in seconds that is used if not overridden in the CMDL mission plan. If this entry is not specified, 60 seconds is used as the default value.

Syntax:

```
commsTimeout = seconds
```

Example:

```
commsTimeout = 300
```

The CMDL Mission Plan File Format