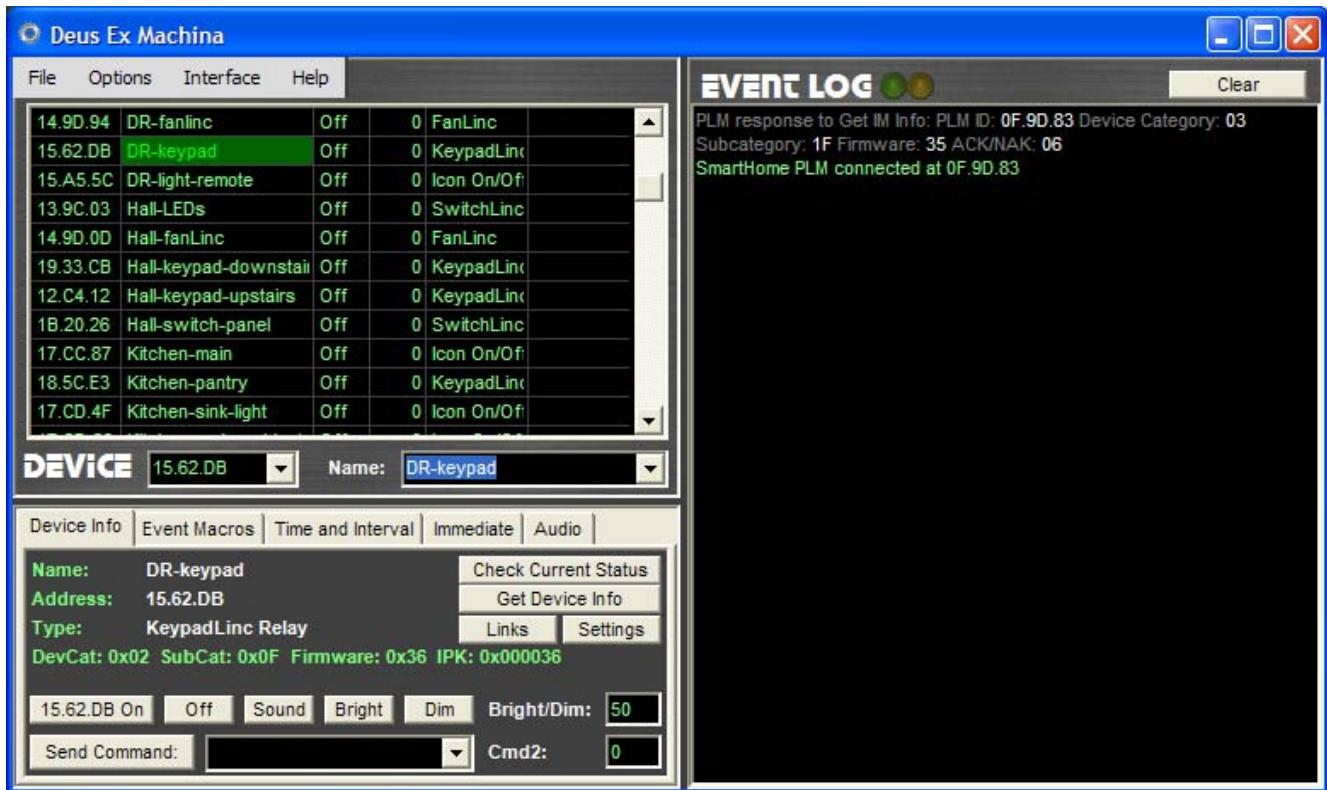


DEUS EX MACHINA

© 2002-2012 JONATHAN DALE
VERSION 3.0.5



Deus ex Machina: 1: a god introduced by means of a crane in ancient Greek and Roman drama to decide the final outcome, 2: a person or thing (as in fiction or drama) that appears or is introduced suddenly and unexpectedly and provides a contrived solution to an apparently insoluble difficulty.

Deus Ex Machina monitors and controls Insteon and X10 devices, as well as sound alerts, effects, and playlists. It was designed originally for interactive theater applications, but is general enough to be used for any Insteon/X10 automation task, including home use. It uses either the CM11A interface, or the Smarthome PowerLine Modem (PLM). The CM11A is sold under the names ActiveHome, Home Director, etc.

The Smarthome PLM has full X10 capabilities and can also communicate with Insteon devices. Sections of the documentation that refer specifically to the PLM or to Insteon devices are outlined like this paragraph for easy reference. The CM11A only supports X10.

Deus Ex Machina tracks and displays Insteon and X10 activity, provides an interface for directly controlling devices, and incorporates its own macro language. The

program must be running for the macros to work; nothing is loaded into the interface's memory.

The display is split into three sections: the Device table provides status information and device selection. The Tab Panel provides control of and specific information about the selected device as well as the ability to create, view, and run macros associated with both events and times, and control of playlists. The Event Log provides an on-going log of all activity sent and received by the program. Additional options are available on the Menu.

A NOTE ON FILES: All user files are saved in

`\My Documents\Deus Ex Machina\ (Windows XP)`

`\Documents\Deus Ex Machina\ (Windows Vista/7)`

Subfolders are also created in that location during installation to store other files needed (e.g. sound effects).

DEVICE TABLE

The grid display shows a list of all devices, with names and current status. From left to right, the columns show the address, name, whether the device is on or off, the current dim/bright setting (0=off, 100=maximum brightness), the type of device, and whether it is being flashed or pulsed (repeatedly dimmed and brightened). Flashing and pulsing are activated by macros; see the Macro language reference for more information. Clicking on a device code or name will select the device; clicking in the on/off column will toggle the device (turning it on if it is off, and vice versa).

Note that the device status will default to “Off” when the program is first started. X10 device status reflects the last command sent to or from the address, as there may be multiple devices at the same address and X10 devices do not send an acknowledgment response (and if the device did not respond, the display may be incorrect).

For Insteon devices, the status is updated after receiving an acknowledgment from the device that the command was received. You can also Check the status of an Insteon device (see below on the Device Info tab panel or use the Check macro command).

ALARM		Off	0	Variable
15.A6.59	Basement-stairs-top	Off	0	Icon On/Of
14.9D.94	DR-fanlinc	Off	0	FanLinc
15.62.DB	DR-keypad	Off	0	KeypadLinc
15.A5.5C	DR-light-remote	Off	0	Icon On/Of
13.9C.03	Hall-LEDs	Off	0	SwitchLinc
14.9D.0D	Hall-fanLinc	Off	0	FanLinc
19.33.CB	Hall-keypad-downstai	Off	0	KeypadLinc
12.C4.12	Hall-keypad-upstairs	Off	0	KeypadLinc
1B.20.26	Hall-switch-panel	Off	0	SwitchLinc
17.CC.87	Kitchen-main	Off	0	Icon On/Of

DEVICE 13.9C.03 Name: Hall-LEDs

Below the grid, the device panel includes drop-down selection of device addresses and names. You can select a device here, or by clicking on the grid display. The Name box also allows the assigning of new names. These names are used in the Event Log and can also be used in Macros. To assign or change a name, select the device by clicking anywhere in its grid row, then type the new name in the name box. To select an already-defined device by name, use the pull-down list. Names may not contain spaces.

To add a device to the table, enter its address in the address box. If the address is a valid X10 or Insteon address, the device will be assigned the appropriate type. Other addresses will be defined as Variables, which can be referenced in macros but do not directly control devices.

TAB PANEL – DEVICE INFO

Below the device table is a tabbed area that allows selection of *Device Info*, *Event Macros*, *Time and Interval Macros*, *Immediate Macros*, and *Audio*.

The screenshot shows the 'Device Info' tab selected. It displays the following information for a device:

- Name: DR-keypad
- Address: 15.62.DB
- Type: KeypadLinc Relay
- DevCat: 0x02 SubCat: 0x0F Firmware: 0x36 IPK: 0x000036

Control buttons include 'Check Current Status', 'Get Device Info', 'Links', and 'Settings'. At the bottom, there are buttons for '15.62.DB On', 'Off', 'Sound', 'Bright', 'Dim', and a 'Bright/Dim' slider set to 50. A 'Send Command' dropdown menu and a 'Cmd2' field with the value 0 are also visible.

The *Device Info* tab provides information about the selected device. For X10 devices, this includes the name and address.

For Insteon devices, this also includes the type of device, device category (DevCat), subcategory (SubCat), firmware version, and Insteon product code (IPK). You can update this information using the *Get Device Info* button. The device type is automatically determined from the device category and subcategory. You can also update the current device status with the *Check Current Status* button.

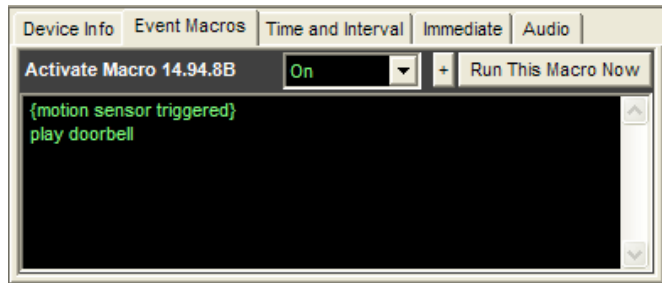
The *Links* and *Settings* buttons get low-level information from the device: the All-Link Database and any Extended Settings respectively. The *Links* button will open a separate window for viewing and editing links, if the device responds. After pressing the Links button, it may take some time for the Links window to load. You should see the green indicator at the top of the Event Log flashing until this is complete. Do not press it a second time. For more information, see the Links section below.

Note that only i2 devices are supported for link editing and viewing.

Below this information are buttons for sending commands to the currently selected device: *On*, *Off*, *Sound*, *Bright*, and *Dim*. Sound is not an X10 command, but plays the sound associated with the device. See the Menu option Sound for more information. (Note that these buttons are disabled when you have selected a Variable.) Next to these is the setting for the *Bright/Dim* value sent with the *Bright/Dim* buttons.

At the bottom of the display, the *Send Command* button will send a command selected from a pull-down list, which provides a way to send special commands such as “all lights off” and “pre-set dim 1”. The command list changes depending on whether you have selected an Insteon or X10 device. The Cmd2 field next to the command is used for other command parameters, depending on the command to be sent.

TAB PANEL – EVENT MACROS



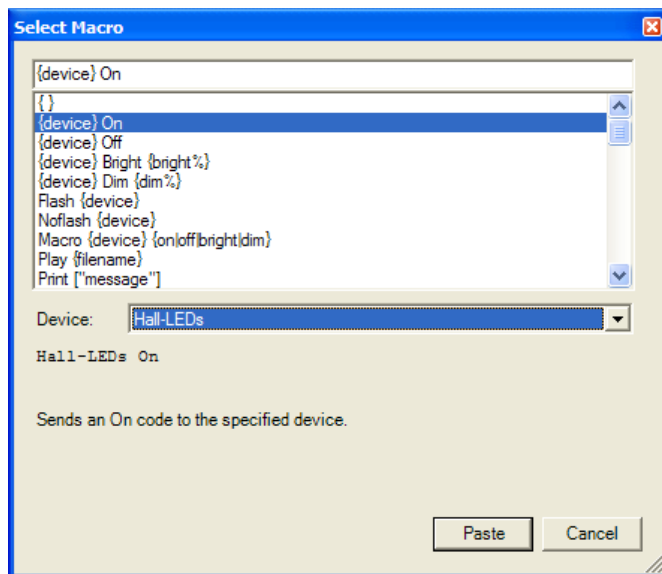
The *Event Macros* tab allows macros to be programmed to run when a signal is received from the device. Macros can be programmed for each device with four events: *On*, *Off*, *Dim*, and *Bright*. The macro for the currently selected device and event is displayed here. When the event is received from the device, the macro will run. See the Programming Macros section for more information about writing macros.

When Dim and Bright commands are sent by holding the button/paddle on an Insteon device, the macro will run repeatedly until the button is released.

If a macro is defined for a device or variable named “startup”, its On macro will run automatically when Deus Ex Machina is started. (This is delayed about 5 seconds in order to allow the connection to the interface to be made.)

The *Run This Macro Now* button starts the macro immediately, and is useful for testing macros.

The *Insert Macro Command* button is labeled simply as +. This button opens a selection window showing all the macro commands and allowing the insertion of a selected command (some also allow selection of parameters, such as the device to be entered.)

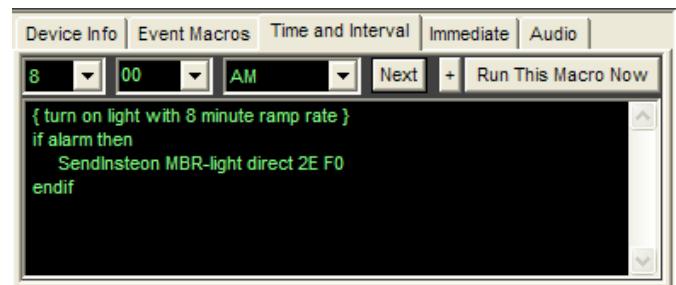


If you make a mistake while entering the macro, pressing Ctrl-Z will restore the macro to what it was when you first

started editing the text. Clicking anywhere outside the Macro box will record your updates. This limited Undo capability is useful for changing mistakes you have just made; if you want to protect yourself from more substantial mistakes, you should save your macros (*File/Save Macro File*).

The *Run This Macro Now*, *Insert Macro Command* +, and ctrl-Z apply to all macro editing, here as well as for time, interval, and immediate macros.

TAB PANEL – TIME AND INTERVAL

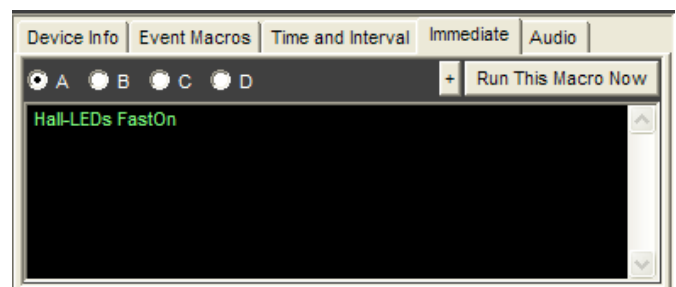


The *Time and Interval Macros* tab lets you define macros that will run at a specific time of day, or at specific intervals. Use the drop-down boxes to select the hour, minute, and AM or PM for your macro. Or use the *Next* button to move to the next time which has an associated macro.

Selecting *Interval* instead of AM/PM allows you to view and edit interval macros. These work just like time macros, except that they happen at intervals of fixed numbers of minutes. For example, the interval macro for 5 minutes will execute once every five minutes.

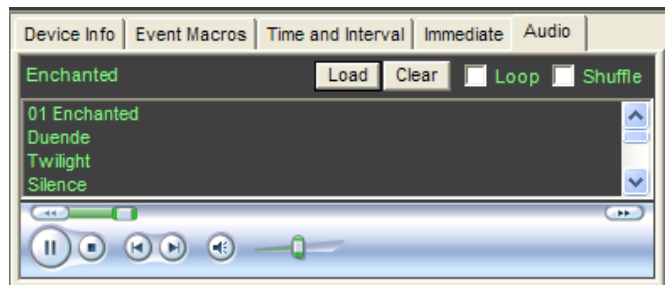
Note that the interval listed as 00 minutes will execute once every 10 seconds. It is not recommended to send device commands this frequently.

TAB PANEL – IMMEDIATE



The *Immediate Macros* tab is a scratch pad for writing and testing macros. It can be used to write macros that will later be copied and pasted to a specific device, or simply used to trigger immediate events. There are four macros here, selected as *A*, *B*, *C*, and *D*. They will be saved along with other macros, but will never be automatically triggered.

TAB PANEL – AUDIO



The Audio panel provides control of a musical playlist.

The **Load** button allows a playlist or music file to be loaded. Playlist files must be Windows Media Player supported playlists (.wpl or .m3u), and can be created and saved in Windows Media Player. Note that they must be used in the location where they were saved, they cannot be copied to a different folder. Loading a playlist will replace the currently loaded playlist (if there is one). Loading a music file (.wav, .mp3, .wma, etc) will add it to the current playlist. You can select multiple files at once and they will all be added to the playlist.

The **Clear** button erases the current playlist, and stops playback.

If **Loop** is checked, when the end of the playlist is reached, it will start over at the beginning. If **Shuffle** is checked, the playlist is played in random order.

The name of the current playing song is shown at the top of the panel, and a list of all the songs in the playlist is shown below. Selecting a song in the list and pressing the **Delete** key will remove it from the playlist.

At the bottom of the tab are the normal Media Player controls (play, pause, stop, previous track, next track, mute, volume, and a position slider).

Playlists can also be loaded and controlled by macro commands.

EVENT LOG

This window shows all device activity. Codes received are listed by device name or by address if the device is not named. Codes sent during macros are prefaced by the macro name (e.g., “Macro A6 On:”, “Macro Motion1 On:”, “Time 12:00 AM:”, or “Interval 05”). The display is color-coded: **times are blue**, **events are green**, **commands sent are yellow**, and **errors parsing your macros are red**. The display can be erased with the **Clear** button. The log keeps the last 30,000 characters of information (about 1000-1500 events).

For extra detail while working on your macros, you can select **Show If/Then, Play** from the Macros menu. This will display extra messages in the event log for **If/Then/Else/EndIf** and **Play** statements. These will appear in red.

If you have enabled the “Show Low-Level Events” option on the menu, PLM events will also be displayed. The names of events and parameters are displayed in dark gray, and the values of parameters are displayed in light gray.



Virtual indicator lights at the top of the Event Log show when messages are being received from devices (green) or sent to devices (yellow). If you are using the W800RF for wireless X10 devices, these are shown by a third light (white).

These indicators are useful while waiting for long processes to finish. When checking the Links for a device, in particular, there can be a significant delay before the Links window opens. During this time the green indicator will flash as each record is received. (If you want to see the details of these transmissions, check the Show Low-Level Events option on the menu under Options.)

MENU COMMANDS

File

Open Macro File allows you to load a saved collection of devices and macros. When the program opens, it automatically looks for a file called "Default.dev" and loads it if available. You can make the program look for a different filename by changing the setting in the Configuration.

Clear Macros erases all loaded macros, while **Clear Macros and Names** also erases all device names.

Save Macro File saves the current collection of macros and names to the same file it was initially loaded from (normally Default.dev).

Save Macro File As allows you to save this information to a different file or location. This will not change the location that Save Macro File saves to.

Options

Sound allows you to turn sound alerts/effects on and off. When Sound is off, no sounds will be produced. When Sound is on, each time an event occurs (whenever an On or Off command is received from any device) the program will attempt to make a sound. It will look for a sound file with the same name as the device (e.g., if the device's name is "lamp", it will look for a file called "lamp.wav") in the Sounds folder, and if present it will play that file. If the device is not named, it will look for the file "undefined.wav". If there is no sound file with the given name, no sound will be made, so you can control which events make sounds and which don't. If multiple sounds are triggered at once, they will automatically be mixed, rather than interrupting each other.

Show If/Then, Play gives you additional feedback on the status of your macros. Normally, only events are displayed in the Event Log. If this option is checked, you'll also see messages about If/Then/Else/EndIf statements and Play statements in your macros.

Show Low-Level Events will show you all activity and status messages from the PLM interface. These can be useful when diagnosing connectivity problems and when experimenting with Insteon, but during normal operation can create a great deal of clutter in your event log.

Sort Devices by Address or **by Name** changes the order in which devices are listed.

Stop All Macros allows you to stop all macros which are currently running. The macros may be triggered again by incoming events.

List Running Macros will list all macros that are currently running. The list will be displayed in red in the Event Log.

Interface

Configuration allows you to set many options, including the selection of your interface and ports. See below for more information.

Clear CM11A Memory erases any information downloaded into the CM11A. Deus Ex Machina does not load anything into the CM11A's memory, but other programs that access the CM11A may do so.

If the program failed to find the CM11A interface, the **Initialize CM11A** command will try again. This will not usually be necessary; the interface will be initialized automatically after selecting the CM11A in the configuration.

Find Insteon Devices will identify all Insteon devices linked to the PLM and add them to the list of displayed devices. Note that the PLM will only receive signals from Insteon transmitters (e.g. remotes, motion sensors, etc) if they are linked to the PLM. Receivers (e.g. ApplianceLincs, LampLincs) do not need to be linked to be controlled by the PLM.

View/Edit PLM Links will open a window listing all links in the PLM link database, and allowing these links to be edited. Note that this only handles the links in the PLM, so you may also need to edit the corresponding links in the device to be linked (use the Links button in the Device Info tab). See the Links section below for more information.

Link PLM as Responder will attempt to link the PLM to another device. To use this menu item, first put the other device in linking mode either by holding down the button on the device or by using the Send Command button to send an Enter Link Mode command (put the Group number in the Cmd2 field, generally this should be 1). Then click the menu item. You will see a confirmation in the event log if successful.

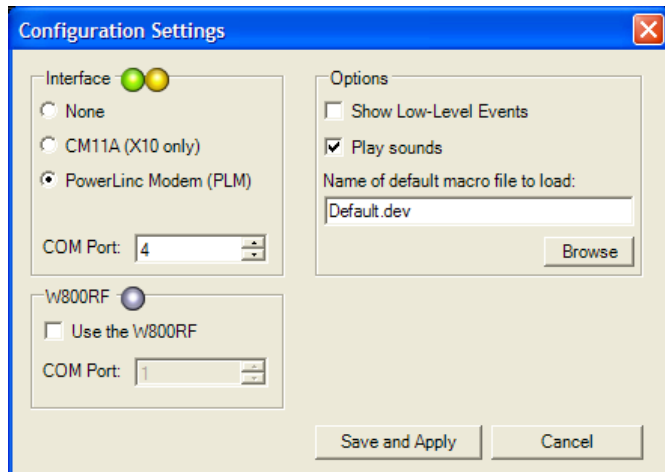
Cancel PLM Linking will take the PLM out of linking mode. (The PLM will also come out of linking mode after successfully linking to another device, or after a short delay.)

Help

About provides basic information about Deus Ex Machina, including the version number.

CONFIGURATION

The configuration window can be accessed from the Menu under Interface. Many system settings can be specified here. All settings are saved to a settings file, “Deus Ex Machina.ini” where they will automatically be loaded the next time you start the program.



The Interface is the device which will be used to communicate with your other devices. When selecting the CM11A or PLM, you must also select a COM port. When using a USB PLM, check your system’s Device Manager under Ports to determine which COM port has been assigned.

The W800RF receives wireless X10 signals directly, rather than requiring them to be received by an X10 wireless receiver and transmitted along the powerline to the CM11A or PLM. Using the W800RF improves the speed and reliability of these devices. If you have a W800RF connected, check the box and select the appropriate COM port (it must be a different port than the CM11A or PLM). The W800RF does not replace the interface, as it does not receive powerline signals (only wireless) and cannot send any type of signal.

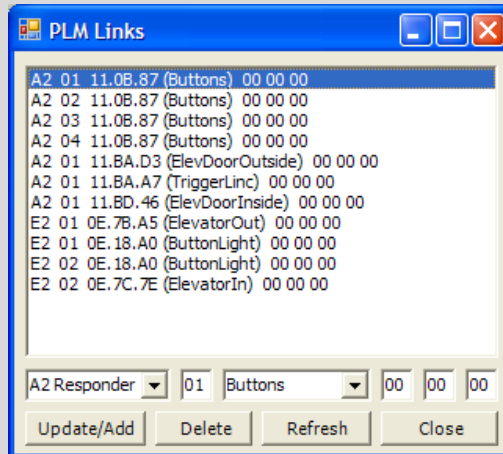
The Options panel allows you to set whether or not you want to see Low-Level Events and whether or not you want Sounds to be played. These settings are identical to the corresponding menu items. You can also change the default macro file that will be loaded when you open the program. You can list either just a filename (in the program folder) or a full file path.

Clicking Save and Apply will save all settings to the INI file. The saved settings will also include whether or not the program is maximized (full-screen), how big the window was, the respective sizes of the panels, and how the devices should be sorted (by address or name).

LINKS

Insteon devices may be addressed directly or by using links to create Groups or Scenes. Links can be created manually on the devices, or edited using Deus Ex Machina. Note that Deus Ex Machina will only edit links for i2 devices which support extended messages.

For manual linking of devices to the PLM, use the *Link PLM* menu item. To edit links in the PLM, use *View/Edit PLM Links* in the menu. To edit the links in a different device, use the Links button on the Device Info tab.



Each line shows the link record Flags, Group, Address (and name if the linked device has one), and three data bytes. For dimmers, the first data byte is the On Level and the second data byte is the Ramp Rate. The third data byte is the Output Group (typically 0, for KeypadLinks it is the button number).

The title bar of the window will show the name of the device whose links are displayed.

To edit a link, click on it, make any desired changes at the bottom of the window, and then click Update. To delete it, click the link and then click Delete. For the PLM, the links list will not be updated until you click Refresh. For other devices, the links that you changed will be updated after they are acknowledged by the device (usually a few seconds).

Note that there is an important difference when editing PLM links versus device links. For PLM links, the existing link will be modified if you keep the same address and group, but a new link will be created if you change either of these. For device links, the selected link will be modified even if everything is changed. The button displays “Update/Add” for PLM links and “Update” for device links, as a reminder.

Also note that selecting “00 Blank” for the Flags field in the link record will not only delete the link, but will also mark it as the last entry in the link database. Additional links after the blanked entry will be ignored.

PROGRAMMING MACROS

Macros are written in a simple programming language. They can send commands, play sounds, wait for periods of time, and make conditional decisions. Macros can be run manually, activated automatically by incoming events, or activated automatically at certain times or intervals.

Formatting conventions

- [] Anything listed in brackets is optional.
- On|Off When two or more terms are separated by a vertical bar | use either word.
- {device} You can enter the address (e.g., "A6") or name (e.g., "Lamp1"). X10 addresses must be listed in the device table to be valid. Variables may also be turned on, off, be brightened or dimmed, and can be used in conditional statements.

Macros are not case sensitive (including device names and addresses). Words in macros are separated by whitespace, which can be spaces or carriage returns. Use these variously to increase the readability of your macros.

Basic Macro Commands

{ [comments] }

Any text enclosed in curly brackets will be treated as a comment, and ignored during macro execution.

Example: {This is a comment}

Macro execution will continue normally after the end bracket.

{device} On|Off

Sends an On or Off code to the specified device. Note that, when sending commands to multiple devices, a delay is recommended. See the Wait command for more information.

Examples: **A1 On**

The On command will be sent to A1.

{device} Dim|Bright {dim %}

Sends a Dim or Bright code to {device}. The {dim %} must be between 0 and 100. Dim and Bright are relative, so Bright 30 will increase the brightness by 30% (but not beyond 100). If {device} is off, this may turn it on, depending on the lamp controller.

Example: RedLamp Bright 30

RedLamp will be brightened by 30%.

{device} [+|-]{dim %}

Similar to Dim and Bright. If + or - are used, the device is brightened or dimmed by the specified amount. If they are omitted, the device is set to the designated brightness level.

Example: RedLamp +30

RedLamp will be brightened by 30%.

Flash {device}

Begin flashing {device} on and off. {device} is toggled once every two seconds.

Example: **Flash A1**

The device A1 will begin flashing.

Noflash {device}

Stop flashing {device}. Depending on the timing, the device may be left on or off.

Macro {device} {On|Off|Bright|Dim}

Activates the macro associated with the {device} event given. Note that if the macro is already running, this command will have no effect – only a single copy of a given macro may be running at any time. (This prevents endless recursive loops and repeat triggering by devices which send frequent signals, like motion sensors.) A message will be displayed in the Event Log to let you know the macro is already running. Also, note that if, for example, macro 1 calls macro 2, further commands in macro 1 will not be executed until macro 2 has completed.

Example: **Macro A1 On**

The macro associated with device A1 being turned On will be run.

Play {filename}

If Sound is on, plays the sound file. It can be a .wav, .mp3, or other sound file. If the extension is omitted, .wav will be assumed. Unless the path is given, it will look for the file in the Sounds folder (located in the program directory). The filename (and path if one is given) cannot have any spaces.

Example: **Play Sound1.wav**

The sound file Sound1.wav will be played, if it exists.

Print ["message"]

Display a message in the Event Log. The message must be enclosed in quotes. If the message is omitted, a blank line will be displayed in the Event Log.

Example: **Print "All systems go."**

The message "All systems go." will be displayed in the Event Log.

Pulse {device}

Begin pulsing {device}. {device} receives a Bright 100 command, followed 4 seconds later by a Dim 100 command; this repeats indefinitely. If {device} is off, this may turn it on, depending on the lamp controller. When using the PLM, it is not recommended that X10 devices be Pulsed, as it will create too much traffic on the power line. Insteon devices may be pulsed without problems.

Example: **Pulse A1**

Device A1 will begin pulsing on and off at 4 second intervals.

Nopulse {device}

Stop pulsing {device}. Depending on the timing, it may be left on or off.

Toggle {device}

If {device} is On, an Off code is sent. If {device} is Off, an On code is sent. Note that for X10 devices, the current setting depends on the last code sent/received, while for Insteon devices the current setting depends on status information from the device.

Example: **Toggle A1**

The device A1 will be turned on or off.

Wait {seconds}

Waits for the specified period of time. This can be useful to allow a command to a device to resolve before sending the next. A 1-second delay can be helpful between Insteon commands. A delay of 1-2 seconds is necessary between X10 commands.

Example: **Wait 10**

Execution of the macro will pause for 10 seconds before continuing.

Waitrandom {seconds}

Waits for a random duration between 0 and {seconds}.

Example: **Waitrandom 10**

Execution of the macro will pause for a period between 0 and 10 seconds before continuing. On average, this will result in a 5 second delay.

If you want to wait for a certain minimum amount of time, use a Wait statement followed by a Waitrandom statement. If you want a non-linear distribution of delays (e.g., a higher likelihood of delays in the middle of the range than at the high or low end), use several Waitrandom statements in a row.

Conditional Macro Commands

If {condition} Then {Command List} [Else {Command List 2}] Endif

If {condition} is true, then {Command List} is executed, otherwise {Command List 2} is executed.

Example: **If A1 On Then A2 On Endif**

If the device A1 has sent or received an On signal more recently than an Off signal, the X10 command On will be sent to device A2. (For Insteon devices, the condition will be evaluated based on the last status or acknowledgment received from the device.)

Example: **If Ubolight Off Then Ubolight On Else Ubolight Off Endif**

If the device named Ubolight is off, it is turned on. Otherwise it is turned off. This is equivalent to the command: Toggle Ubolight

Note that it might be easier to read an If/Then command if you space it out:

```
If Ubolight Off Then
    Ubolight On
Else
    Ubolight Off
Endif
```

If/Then statements may be nested to any level. For example, both of the following examples will work:

```
If A1 On Then
    If A2 On Then A3 On Else A3 Off Endif
    If A4 On Then A5 Off Else A5 On Endif
Else
    If A2 On Then A3 Off Else A3 On Endif
Endif
```

```
If A1 On Then
    If A2 On Then
        If A3 On Then
            A4 On
        Endif
    Endif
Endif
```

{condition} may take any of several forms:

{device} [On|Off]

True if {device} is on. If “On” and “Off” are omitted, the condition will default to “On”.

{device} <|> {Num}

True if the current brightness for {device} is greater/less than {Num}.

Example: **If Ubolight > 50 Then Ubolight Dim 100 Else Ubolight Bright 100 Endif**

If the device named Ubolight is set to at least 50% brightness, it is dimmed by 100% (so, down to 0%). Otherwise it is brightened by 100%. This is equivalent to one iteration of the Pulse condition.

Note that the brightness for X10 devices is computed from bright and dim commands sent and received for the device, not actually received from the device. The program has no information about the state of X10 devices when it begins, so errors are possible. For Insteon devices, it is based on the last status or acknowledgment received.

Random

There is a 50% chance that Random will be treated as True, and 50% chance it will be False.

Example: **If Random Then Play Sound1 Else Play Sound2 Endif**

Either Sound1.Wav or Sound2.Wav will be played, assuming the files exist and sound is turned on under the Options menu.

Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday

The condition is true if it is currently the specified day of the week.

Example: **If Monday then play theblues endif**

If today is Monday, play theblues.wav.

Weekday | Weekend

Weekday is true Monday-Friday, and Weekend is true Saturday and Sunday.

Example: **If Weekday then Lights On endif**

If today is a week day, turn on the device Lights.

And

Multiple conditions can be combined with And to require that they must all be true.

Example: **If Lamp1 on and Lamp2 off Then Play Sound1 Endif**

Sound1 will only be played if Lamp1 is on and Lamp2 is off.

Or

Multiple conditions can be combined with Or to execute a condition if any one of them is on. You can also combine And with Or; the expression is evaluated from left to right.

Example: **If Lamp1 or Lamp2 or Lamp3 Then Play Sound1 Endif**

Sound1 will be played if Lamp1, Lamp2, or Lamp3 is on.

Example: **If Alarm and Weekday then Lights Slowon Play Wakeup endif**

If the device Alarm is on (this could be a variable used to conveniently disable your automated alarm clock) and today is a weekday, turn the lights on slowly and play Wakeup.wav.

When using Insteon devices in If/Then statements, the If statement will use the current status of the device, as most recently reported by that device. This improves accuracy and reliability as compared to X10 devices, for which the status is inferred from commands sent and received, but without direct feedback from the device. This applies to both On/Off status as well as brightness level.

However, note that the status will not be updated during execution of your macro. All status updates from devices are processed only after the macro has completed. The exceptions are during the execution of Wait and Waitrandom statements; status updates may be processed while your macro executes these statements.

For example, with X10 devices you could use a macro like:

```
If Lamp1 On Then Lamp1 Off Else Lamp1 On Endif  
If Lamp1 On Then Lamp1 Off Else Lamp1 On Endif
```

After the first If statement, the status of Lamp1 would be updated. The result would be Lamp1 briefly turning on and then off (or vice versa, depending on its original state).

If Lamp1 was an Insteon device, the status of Lamp1 would not be updated after the first If statement, so if, for example, the first If statement turned Lamp1 on, then the second If statement would also try to turn Lamp1 on. Adding a Wait statement between the two If statements could correct the problem, if it was a long enough wait for the device to report its status (typically less than a second, but could be longer if powerline conditions are not optimal).

In this particular example, a better solution would be:

```
If Lamp1 On Then
    Lamp1 Off Lamp1 On
Else
    Lamp1 On Lamp1 Off
Endif
```

Playlist Commands

Playlist {filename}

Load a playlist (.wpl, .m3u) or song (.mp3, .wma, etc). It will begin playing immediately. If a playlist is loaded, it will replace the current playlist. If a song is loaded, it will be added to the current playlist. If no path is given, the Sounds folder will be assumed. If there are any spaces in the filename or path, it must be enclosed in quotes. If you use the Insert Macro Command button (+) to insert this macro command, you can browse to your file and have the filename and path added for you.

Example: **Playlist "C:\Documents and Settings\Username\My Documents\My Music\
 My Playlists\playlist.wpl"**

Load the specified playlist and begin playing it immediately.

Playlist clear

Clear the current playlist, and stop playing.

Playlist play

Playlist start

Either of these will start playing the current playlist.

Playlist pause

Pause the current playlist. If it is resumed, it will do so at the same point.

Playlist stop

Stop playing the current playlist. If it is resumed, it will do so from the beginning.

Playlist next|previous

Skip to the next or previous track.

Playlist shuffle|noshuffle

Switch between playing in random order (shuffle) or in playlist order (noshuffle).

Playlist loop|noloop

When loop is selected and the end of the playlist is reached, it will begin again at the beginning. If noloop is selected, it will stop.

Playlist remove

Remove the currently playing song from the playlist. If no song is playing, it has no effect.

Playlist volume [+|-]{number}

Volume must be a value 0-100. If + or – are used, the volume is increased or decreased by the specified amount. Otherwise, volume is changed to the specified level.

Playlist volume {device}

Set the volume to the same value as the Level (brightness) of the specified device or variable.

Insteon Macro Commands

These macro commands use advanced features of the PLM or of Insteon devices.

{device} FastOn|FastOff

Normally, Insteon light controllers will turn on lights gradually at a speed determined by the ramp rate for the controller. FastOn causes them to turn on instantly at 100%, and FastOff turns them off instantly. If used on an X10 device, this is treated as a normal On command.

Example: **Lamp1 FastOn**

Turns on Lamp1 abruptly.

{device} SlowOn|SlowOff

Lights can also be turned on slowly. This uses the Light On at Ramp Rate command to turn the light on to 100%, but takes 19 seconds to reach full brightness. This command is not supported on all devices. For example, it only works on LampLincs starting with version 4.1 (firmware 0x32). If used on an X10 device, this is treated as a normal On command. SlowOff turns the device off using Light Off at Ramp Rate.

Example: **Lamp1 SlowOn**

Turns on Lamp1 slowly.

Check {device}

Requests the status of the specified Insteon device. The macro then proceeds without waiting for a result. The status will be updated when the macro completes, or if a Wait or Waitrandom command is executed.

Example: **Check Lamp1**

Command {command}

Device-triggered macros can be activated by multiple types of event. For example, the On macro can be triggered by On or Fast On. This command allows you to specify precisely which command must have been received. Anything after this command will be executed only if the specified command is the one that triggered the macro. To resume execution, use Command All. The commands are:

On, Off	The macro was triggered by a normal On or Off event.
FastOn, FastOff	Triggered by a Fast On or Fast Off event (generally from double-tapping.)
OnAtRate, OffAtRate	Triggered by an On At Rate or Off At Rate event.
Bright, Dim	Triggered by a Bright or Dim event (X10 or Insteon).
ManualChange	The Manual Change event occurs when you dim/brighten an Insteon device by holding the paddle or button.
Click	Triggered by clicking the Run This Macro Now button within Deus Ex Machina.
Startup	The macro ran automatically when Deus Ex Machina started up. (Only for the On macro associated with a device named Startup.)
All	Resume execution of the remaining part of the macro regardless of how it was triggered.

Example:

```
Command ManualChange
Playlist Volume +1
```

While the paddle/button is being held down, increment the playlist volume by small steps. This allows direct and immediate control of the playlist volume using a linked Insteon device. You would use this command in the Bright macro and a similar command with Volume -1 on the Dim event. If the Bright event may also be received (typically only from a device with a dedicated Bright button, such as a RemoteLinc, or an X10 device), you could use it to increment the volume in larger steps.

Group {Num|All}

Most Insteon messages are directed at groups. For devices that can send multiple signals, the group number may indicate the type of signal. For example, the RemoteLinc uses group 1-6 for its six buttons, and the KeypadLinc uses groups 1-8. If button 1 is pressed, the signal will be sent to group 1. This macro command allows you to write code that responds to a specific button press. When this command is executed, the macro will only execute the subsequent commands if the triggering event was the specified group. Otherwise, the macro will skip everything before the next Group statement. To execute code for all groups, use Group All. X10 events and macros called with the Macro command are always treated as Group 0.

Example:

```
Group 1 Lamp1 On
Group 2 Lamp2 On
Group All Play Sound1
```

If this macro was triggered by Group 1, turn on Lamp1. If it was triggered by Group 2, turn on Lamp2. Regardless of the group, play Sound1.

Scene {Num} {On|Off|Faston|Fastoff}

Tells the PLM to send a command to a group of devices (also known as a scene) that have been linked as responders to the PLM. The PLM will send a group broadcast command, followed by cleanup direct messages to each linked device in the group. Sending other Insteon or X10 commands will interrupt the cleanup messages, so it may be helpful to add a Wait command after Scene. Scene commands can be faster and more reliable than direct commands. The command "Link" can be used interchangeably with "Scene".

Example: **Scene 1 On**

Sends a group broadcast telling linked group 1 to turn on, and then follows it with cleanup direct messages to each device in the group.

SendInsteon {device} {{flags}|direct|broadcast|group} {cmd1} {cmd2}

Send a standard 9-byte Insteon command. The command is sent from the PLM to the specified device.

The Flags parameter determines whether the message is a broadcast or direct message, the number of hops, and whether it is an acknowledgment. You can specify an actual value in hex, or just enter Direct (0F), Broadcast (8F), or Group for a group broadcast message (CF).

Cmd1 is the command; these are the same commands listed in the Send Command pulldown list for Insteon devices. The command number must be in hexadecimal. (The Send Command list in the Device Info tab gives the hex value for each command.)

Cmd2 gives additional information such as the bright/dim level (on a scale of 00-FF) or the group number. This command must also be in hexadecimal.

Example: **SendInsteon Lamp1 direct 11 FF**

Tells Lamp1 to turn On (command 11) to 100% (FF), as a direct message (0F).

Sendextended {device} "{bytes}"

Sends an extended insteon message to the specified device. The flag byte is automatically set to 1F, it does not need to be set here. The {bytes} should be hexadecimal numbers separated by spaces, and must be enclosed in quotes. They should begin with command1, command2, and then 14 data bytes. If the message is too short, 00 will be filled in for any remaining bytes.

Example: **Sendextended keypadlinc "2E 00 00 09 40"**

Sends the extended command 2E (Extended Set/Get) to a device named "keypadlinc." This particular example will turn on buttons 4 and 6, and turn off all other buttons except those connected to the load. The remainder of the command is 00's, so it has been omitted here.

Sendbytes "{bytes}"

Sends raw data to the PLM. It can be any PLM message. The {bytes} should be hexadecimal numbers separated by spaces, and must be enclosed in quotes.

Example: **Sendbytes "02 6B 40"**

Sends the PLM command 02 6B 40, which puts the PLM in Monitor Mode. (In this mode, the PLM will receive any message from any device which is linked to the PLM, whether or not the message is addressed to the PLM. Normally, the PLM will only receive messages which are addressed to it.)