Media Pro® 4000

Operation Manual &

Programming Reference

Version 2.06

Release Date 7.31.01

Document Revision 2.06

Release Date 7.31.01

Great effort has been made to provide accurate and current information within this manual, although, errors occasionally go undetected. If you discover an error, please bring it to our attention so we may correct it. Anitech Systems, Inc. will not be held responsible for any inaccuracies or omissions, or any consequences resulting from them.

Anitech Systems, Inc. reserves the right to make revisions to its hardware and software products without notice.

Anitech Systems, Inc. assumes no responsibility or liability for the use of any of these products and makes no representation, expressly or implied that the use of its products will be suitable to all applications.

The Media Pro® 4000 system in not intended for direct control in safety critical applications. It should be used in conjunction with a Programmable Logic Controller where safety is an issue.

The data in this document incorporates proprietary rights of

Anitech Systems Inc.
25021 Anza Drive, Valencia, CA 91355 USA
Phone (661) 257-2184 Fax (661) 257-2025
Email - Mail@Anitech-Systems.com
Web - http://www.Anitech-Systems.com

Contents

Preface

Welcome to the Media Pro® User's Operational Manual. Highlighted in this manual are the configuration, software operation, programming reference, examples of typical applications, and technical support information.

This document is provided so users of the Media Pro® 4000 can gain an understanding of the system design and implementation techniques. It is assumed the reader has a basic knowledge of typical show elements and their control requirements. In addition, a basic understanding of programming concepts will help the user understand Anitech System's powerful English based Media Pro® Control Language (MPCL).

For specific information about other Media Pro® 4000 products, please refer to the applicable user manual or on-line help system.

Related Publications

The following documents contain additional information concerning ASI Media Pro® 4000 products. To obtain a copy of any of the documents listed below, contact ASI or visit our website.

Document	Description
ABM – 4010	Allen Bradley Remote Input/Output Module (RIO)
AOM – 4010	Analog Output Module - 16 Channel
APC – 4020	Animation Programming Console
ASM – 4030/4020	Analog Servo Module - 8 Channel / 4 Channel with Compliance
DSM – 4020	Digital Sound Module with PCMCIA socket
HMR – 4000	Horizontal Module Rack - 5 Slot
ICM – 4020	Intelligent Control Module
IMC – 4020/4010	Integrated Module Controller
IOM – 4020	Input/Output Module - 16 Bit (24vdc)
LCM – 4020	Lighting Control Module (DMX 512)
SEM – 4020	Serial Expansion Module - 8 Channel (RS-232)
TCM – 4020	Time Code Module (SMPTE)
VMR – 4000	Vertical Module Rack - 18 Slot

System Requirements

- IBM PC compatible computer running Windows 95 / 98 environment with 1 serial port at 19.2Kbps.
- The PC should be at least Pentium 90 MHz, with a minimum of 16 Mbytes of Ram.
- The recommended minimum system is Pentium 400 MHz or faster with 32M RAM and a serial port at 115.2Kbps.
- An EPP 1284-1994 compatible parallel port is required for animation and is recommended for all other operations, but not required. Downloads for Animation and DSM sound files are much faster over the EPP parallel port.
- Approximately 12 Mbytes of disk space are required for the MP4000/APU4000 software. In addition, we recommend a minimum of 30 Mbytes storage for each show's application files. This size will vary depending on the show configuration and the size of the sound and animation files that may be required.

Getting Started

The Media Pro® 4000 Software is provided on a set of floppy disks. To install the files, follow these directions:

- ☐ Place disk 1 into the floppy drive. Open the Explorer and browse **FLOPPY [A:]**
- ☐ Double click on **SETUP… .EXE**
- ☐ Follow the on-screen installation instructions.
- ☐ The default directory **C:\MP4000** will be created and placed on the desktop.
- ☐ To enter the program, browse the directory and double click **MP4000.EXE**
- ☐ Current versions of the MP4000 software place an icon on the desktop, earlier versions (1998) do not.

Software License Agreement

ANITECH SYSTEMS MEDIA PRO 4000 SOFTWARE RELATED COMPONENTS,
INCLUDING, BUT NOT LIMITED TO MEDIA PRO 4000 SYSTEM SOFTWARE
and ANIMATION PROGRAMMING UTILITY.

END-USER LICENSE AGREEMENT FOR ANITECH SYSTEMS SOFTWARE PRODUCT

IMPORTANT-READ CAREFULLY:
This Anitech Systems End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Anitech Systems for the Anitech Systems software product(s) identified above which may include associated media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT").
By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT.

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as, other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE. The SOFTWARE PRODUCT is licensed as follows:

* Installation and Use. Anitech Systems grants you the right to install and use copies of the SOFTWARE PRODUCT on your computers.

* Backup Copies. You may also make copies of the SOFTWARE PRODUCT as may be necessary for backup and archival purposes.

2. OTHER RIGHTS AND LIMITATIONS.

* Pre-release Software. If any component of the SOFTWARE PRODUCT or any of its components is marked "Prerelease" or "Beta", the component of the SOFTWARE PRODUCT constitutes pre-release code and may be changed substantially before commercial release.
You may not use such component in a live operating environment where it may be relied upon to perform in the same manner as a commercially released product or with data that has not been sufficiently backed up.

* Maintenance of Copyright Notices. You must not remove or alter any copyright notices on all copies of the SOFTWARE PRODUCT.

* Distribution. You may not distribute copies of the SOFTWARE PRODUCT to third parties.

* Prohibition on Reverse Engineering, De-compilation, and Disassembly. You may not reverse engineer, de-compile, or disassemble the SOFTWARE PRODUCT.

* Rental. You may not rent, lease, or lend the SOFTWARE PRODUCT.

* Upgrade/Update. If this Software is an upgrade from a previous version of an Anitech Systems product, you may not transfer the prior version to another user.

* Transfer. You may permanently transfer all of your rights to use the Software on a permanent basis under this EULA, provided the recipient agrees to the terms of this EULA.

1) You transfer this Agreement, the Software, and all accompanying written materials and retain no copies;
2) The recipient agrees to the terms of this Agreement;
3) You first notify Anitech Systems in writing of the date of transfer and the recipient's full name and address.

This Agreement will automatically terminate without notice to you if you fail to comply with its terms.

* Support Services. Anitech Systems may provide you with support services related to the SOFTWARE PRODUCT ("Support Services").
Any supplemental software code provided to you as part of the Support
Services shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this EULA. With respect to technical information you provide to Anitech Systems as part of the Support Services, Anitech Systems may use such information for its business purposes, including for product support and development.

* Compliance with Applicable Laws. You must comply with all applicable laws regarding use of the SOFTWARE PRODUCT.

3. TERMINATION. Without prejudice to any other rights, Anitech Systems may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT.

4. COPYRIGHT. All title, including but not limited to copyrights, in and to the SOFTWARE PRODUCT and any copies thereof are owned by Anitech Systems. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by Anitech Systems.

5. LIMITATION OF LIABILITY. To the maximum extent permitted by applicable law, in no event shall Anitech Systems or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the SOFTWARE PRODUCT or the provision of or failure to provide Support Services, even if Anitech Systems has been advised of the possibility of such damages.

Anitech Systems expressly disclaims all warranties, express and implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

Anitech Systems does not warrant that the functions contained in the software will meet your requirements, or that the operation of the software will be uninterrupted or error-free, or that defects in the software will be corrected. The entire risk as to the results and performance of the software is assumed by the End-User.

In any case, Anitech Systems' entire liability under any provision of this EULA shall be limited to the amount actually paid by you for the SOFTWARE PRODUCT.

* Examples. Examples & samples are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Anitech Systems cannot assume responsibility, or liability for actual use based on examples.

6. MISCELLANEOUS.

This EULA is governed by the laws of the State of California, U.S.A.

Should you have any questions concerning this EULA, or if you desire to contact Anitech Systems for any reason, please contact:

Anitech Systems, 25021 Anza Drive, Valencia, CA 91355;
Telephone - (661) 257-2184
FAX - (661)257-2025
Email - Admin@Anitech-Systems.com
Web site - www.Anitech-Systems.com

Physical Dimensions

These are the approximate physical dimensions and space recommendations for the Media Pro® 4000 units.

1 - VMR-4000 Approximate Dimensions
- A) 19" EIA Standard Retma Rack Mount Package
  - 1) Width 19"
  - 2) Height 10.5"
- B) Overall Depth 8.5"
  - 1) 7.75" Behind Rack Mount Surface
    - (a) 1.5" Minimum Additional Depth Necessary for Cables
  - 2) 0.75" In Front of Rack Mount Surface
  - 3) 10" In Front of Panel When Using Panel Mount Sides
- C) Recommended - 1 Rack Space (1.75") Vent Panel - Above, Below, and Between Racks

2 - HMR-4000 Approximate Dimensions
- A) 19" EIA Standard Retma Rack Mount Package
  - 1) Width 19"
  - 2) Height 5.25"
- B) Overall Depth 8.5"
  - 1) 7.75" Behind Rack Mount Surface
    - (a) 1.5" Minimum Additional Depth Necessary for Cables
  - 2) 0.75" In Front of Rack Mount Surface
  - 3) 10" In Front of Panel When Using Panel Mount Sides
- C) Recommended - 1 Rack Space (1.75") Vent Panel - Above, Below, and Between Racks

3 - IMC-4020 Approximate Dimensions
- A) 19" EIA Standard Retma Rack Mount Package
  - 1) Width 19"
  - 2) Height 1.75"
- B) Overall Depth 8.5"
  - 1) 7.75" Behind Rack Mount Surface
    - (a) 2" Minimum Additional Depth Necessary for Cables
  - 2) 0.75" In Front of Rack Mount Surface
- C) Recommended - 1 Rack Space (1.75") Vent Panel - Above, Below, and Between Racks

4 - IMC-4010 Approximate Dimensions
- A) Table Top Package
  - 1) Width 7.5"
  - 2) Height 1.75"
- B) Overall Depth 8.5"
  - 1) 2" Minimum Additional Depth Necessary Behind IMC for Cables

5 - Modules Approximate Dimensions
- A) Eurocard Modular Design
  - 1) Width 0.8" (1 Slot)
  - 2) Height 10.25"
  - 3) Depth 7.5"
    - (a) 0.75" In Front of Rack Mount Surface

6 - PSM-4020 Power Supply Module (Internal Logic) Approximate Dimensions
- A) Included in VMR-4000
  - 1) Width 2.4" (occupies dedicated PSM slot in VMR-4000)
  - 2) Height 10.25"
  - 3) Depth 7.5"

7 - APS-40ES30 Power Supply (Internal Logic) Approximate Dimensions
- A) Included with HMR-4000, IMC-4010/20, APC-4010/20
- B) Table Top Package
  - 1) Width 3.25"
  - 2) Height 2.5"
  - 3) Depth 6.75"
    - (a) 2" Minimum Additional Depth Necessary for Cables

Operating Environment

All Media Pro® Products are manufactured to the highest standards. With proper care and maintenance, they should provide many years of trouble-free service.

To ensure that your equipment has the longest life possible, it should be placed in an area with good ventilation and low humidity, out of direct sunlight and away from heat sources or lamps. Never expose equipment to moisture of any sort. Always maintain a dust, dirt and smoke free environment.

Always remember that high temperature is the enemy of all electronic equipment.

Environmental Conditions:

Recommended Operating:	10° to 32° C (40° - 90° F)
Storage Temperature:	-40° to 60° C (-4° - 140° F)
Relative Humidity:	0 to 95% (Without Condensation)

Warranty

Anitech Systems warrants this product to be free of manufacturing defects for 1 year from the date of purchase. At Anitech System's discretion, Anitech Systems will repair or replace a module that fails due to manufacturing defects.

The warranty does not cover shipping charges or modules damaged due to improper configuration, misapplication, misuse, abuse, accidents, or shipping damage.

Service Options

Obtain a Return Materials Authorization by contacting Anitech Systems.
The contact information for customer support follows:

E-mail	Mail@Anitech-Systems.com
Web	http://www.Anitech-Systems.com
Telephone	(661)257-2184
Fax	(661)257-2025

Non-Warranty Modules Returned for Repair-

Charges will be based on parts used, labor, and shipping charges. Make sure the product is properly packed and insured. Anitech Systems is not responsible for damage that occurs during shipment.

1 Section 1 – System Introduction

The Media Pro® 4000 is a Media Processing Control System for show venues. It consists of a family of full height Eurocard modules and software, which are used to create and control numerous show elements. Devices such as Laser Disk Players, CD Players, Video Switchers, Digital Audio devices, MIDI devices, DMX Dimmers/fixtures, Programmable Logic Controllers, Motion Bases and Animated Characters are among the many elements that can be controlled by the system. [In addition, show resources such as the Solid State Audio/Video and Operator Information can be created and integrated into the same system.] All these "show resources" are available to the designer using a common Media Pro® Control Language throughout the system.

The Media Pro® 4000 architecture allows it to be used in applications that range from a simple single-rack solution controlling a few serial devices, to a distributed multi-rack show control system. A 1-Megabit Media Pro® Network integrated into each rack processor provides communication between system racks. The modular design of the Media Pro® 4000 permits solution for small show venues with just a few modules in a single rack. In larger shows, the solution may entail multiple racks, or distribution across disciplines – but still using the same modular approach.

- Section 1.0 reviews the Media Pro® 4000 System Architecture.
- Section 2.0 is a tutorial and overview of show control systems concept and usage.
- Section 3.0 details the MPCL language reference format.
- Section 4.0 presents how the Media Pro® 4000 modules are programmed with examples
- Section 5.0 gives technical support information and answers some Frequently Asked Questions.

1.1 Purpose and Intended Use

These products are intended for control and integration of theater-style shows and animated rides.

Not to be used as the sole controller in Safety Critical Situations! Should be integrated with a Programmable Logic Controller (PLC) to share information via remote I/O, DF1, or direct connection.

The Media Pro® 4000 has been used to implement a variety of exhibits throughout the world. A partial list of projects around the world using Media Pro® controllers follows:

- Crown Casino, Australia
  - ★ Atrium
- Disney Epcot Center, Florida
  - ★ Millennium Fireworks Display
  - ★ Body Wars
  - ★ General Motors Exhibit
  - ★ SMRT –1
  - ★ AT&T Einsel Theatre
- Disney MGM Studios
  - ★ Star Tours
- Disneyland, California
  - ★ Star Tours
- Expo 86, Vancouver, BC
  - ★ Canada Place
  - ★ BC Main Theater
  - ★ Subscan
- Fox Studios, Sidney, Australia
- Knotts Berry Farm, Anaheim
  - ★ Kingdom of Dinosaurs
- Las Vegas
  - ★ Excalibur, Merlin & Dragon Control System
  - ★ Hilton Hotel, Race & Sports Book
  - ★ MGM Grand, Wizard of Oz
  - ★ Venetian Hotel
  - ★ Frontier Casino, Remote Antenna Controller
- Movie World, Germany
  - ★ Batman Adventure
  - ★ 4th Dimension Theater
  - ★ Police Academy
  - ★ Wild West Animal Stunt Show
  - ★ Looney Tunes
  - ★ Movie Magic
  - ★ Bermuda Triangle
  - ★ Maverick Stunt Show
  - ★ Lethal Weapon
  - ★ German Film History Theater
  - ★ Never-Ending Story
  - ★ WB Classics
  - ★ Golden Garter
  - ★ Cartoon Theater
- Museum of Science and Industry, Los Angeles & Chicago
  - ★ GM Driver Lab
  - ★ GM Wizard of Change
  - ★ Crossroads (Chicago only)
- Sony Metrion, San Francisco
  - ★ PODs
  - ★ Wings
- Universal Studios, Hollywood
  - ★ Back to the Future
  - ★ Star Trek
  - ★ Backdraft
  - ★ King Kong
  - ★ Conan, the Barbarian
  - ★ Miami Vice
  - ★ Collapsing Bridge
  - ★ Earthquake
- Universal Studios, Florida
  - ★ Murder She Wrote
  - ★ King Kong
  - ★ Hitchcock Theater
  - ★ Animal Actors
  - ★ Screen Test Theater
  - ★ Back to the Future
- Universal Studios, Japan
  - ★ Park wide standard show control

1.2 Media Pro® 4000 System Architecture

This section introduces the system architecture and configuration. Please refer to Section 3.0 for system application examples.

During the development of the Media Pro® 4000 we defined 4 major goals:

- Modular, Scaleable, and Extensible Implementation.
- Integration of Event and Real-Time control techniques.
- Direct Connectivity to Industry Standard Programmable Logic Controllers.
- Provide a Common Media Pro® Control Language (MPCL) for all system resources.

1.3 Modular, Scaleable, and Extensible Architecture

The system is comprised of *modules* and racks dedicated to perform specific functions as required by each application. Modularity provides a system with distributed power, and thereby distributed cost, only as required. For simple applications only a few modules and a five-(5) slot rack are required, providing a very powerful and economic solution in many cases. But as systems need to expand, the investment made in the modules can be reused and maintained across a wide range of applications. Consequently, it provides the user with a one-time training and spares inventory that can be amortized over all the needs of a location.

For instance, the ICM or Intelligent Controller Module provides the control, memory, and communications functions for the system. Since each rack requires an ICM, as a system expands and incorporates more modules, the communication and control functions are provided to meet the expansion. The ICM includes a built-in 1-megabit isolated network. As the system is *scaled* up, the additional racks just plug into the network and become part of the application.

Each module contains personality information that is stored in non-volatile Flash Memory on each unit. This memory is divided into either two or three sections: *Module Based Code*, *Parameters* and *Show Specific Data* (if applicable). This feature allows for easy field system *extensibility*.

The *Module Based Code* memory is that portion of executable software contained in the on-board processor for the specific module. This software dictates the module functionality and can be easily upgraded in the field via a PC or Laptop using MP4000, the Media Pro® 4000 user interface and programming tool. This means as code improvements or functional enhancements are released they can be upgraded on-site by the user; the module does not have to be removed and returned to the factory. System updates are available via the Internet.

The *Parameters Based Data* memory is used to store user settings of options provided by the specific module. For a DSM (Digital Sound Module) the parameters for track rate, looping options, etc. are all stored in this memory. The ASM (Analog Servo Module) stores the servo channel gains, accelerations and velocity parameters for each of the channels the board supports. All of these parameters are set by MP4000, the Media Pro® Software, and downloaded from a PC or Laptop.

The *Specific Show Data* memory stores an actual application resource. These resources are controlled just like external resources, such as a video laser disk. Resources are played, stopped, stepped, etc. Examples of these resources are the digital audio data stored in the DSM, the figure animation data stored in the ICM, and the lighting control cues stored in the LCM. In all cases, the control of these devices is done with the same Media Pro® Control Language (MPCL) used in the Media Pro® Cues.

To illustrate this concept, a Laser Disk Player is a typical object. It contains media resource material in the form of video frames. To use these frames in a show, we need to be able to execute a few basic commands to the player. We need to seek the player to a video frame, play the video for a certain number of frames and then stop the video. These basic commands are usable on most of the devices to be controlled in a given venue. The commands are the same for a CD player, a Solid State Audio player, or an Animation Data File.

The advantage of this object concept is -- whatever device is being controlled, the MPCL statements are basically the same.

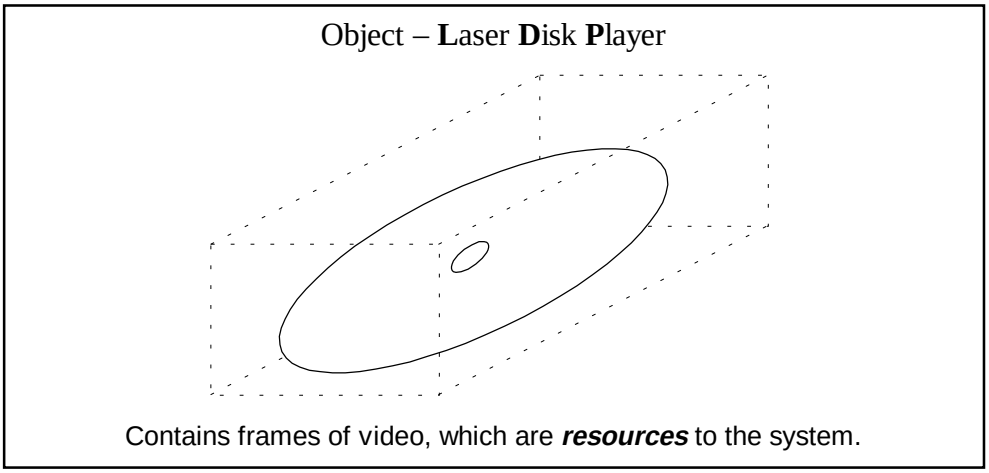


Figure 1 - 1

1.4 Event and Real-Time Techniques

Either *Event* or *Real-Time* programming techniques can solve most of the applications we see today. These two solutions are extremely different and each has distinct advantages over the other depending on the application. Both are available simultaneously in the Media Pro® 4000 and can be dynamically and cooperatively used as required.

> ***Event*** programming allows a designer to script events based on internal clock or SMPTE time code. The events can include playing a laser disk player, triggering a solid state audio sound, ramping DMX controlled light channels or initiating playback of animation data. The events of most devices can be classified and contained in a common control language. This allows the designer to control these devices without intimate knowledge of the device being controlled. A list of these Events is contained in a Cue and can be triggered by external devices such as sensors or PLC connections. In section 3, the Media Pro® Control Language (MPCL) programming techniques are discussed.
>
> ***Real-Time*** programming is accomplished by using an animation programming console that contains switches and proportional potentiometers allowing an animator to record data into the system in real-time. This is typically used for animated characters or motion base applications. In addition, lighting, audio panning, etc. can all be accomplished in this manner. These types of applications are typically done at 24 or 30 frames per second depending on the synchronization requirements of the show.

Real-Time techniques provide high-speed, high-volume data with the trade-off being larger memory storage requirements. Where Event techniques would only store the data for each event that is required, Real-Time recording will store data for each channel that it is running. However, for devices such as animation and motion bases this is required to provide the life-like responses expected from the application. It also provides a much more efficient means to record and store the large amount of data rather than scripting each frame one-by-one.

As you can see, both of these techniques are useful and can be used to accomplish all facets of show control. The real power comes when they can be combined and used together to solve a myriad of show control challenges. The applications described in Section 3.0 will convey how the two techniques are used to provide this powerful solution.

Internally, the Media Pro® 4000 System creates a clock source for system timing and synchronization. The *Internal Clock* provides 30 hertz frame rate derived form an extremely accurate crystal-controlled source. This clock is then used to derive 29.97, 25, 24, 15, and 12 frames/second rates for assigned cue and animation playback control.

If, however, an external video sync source is connected to the video input of the ICM, all the internal clocks become ‘slaves’ to that external video sync source. External sync rates for NTSC, HIDEF, and PAL are accommodated. The locking of internal clocks to external video rates permits easy synchronization of animation data played at 30 frames/second to video material played at 29.97 frames/second. In most cases this eliminates the need to use an audio channel for SMPTE on the laser disk.

1.5 Direct Connectivity to PLC

Since so many of the applications today contain a Programmable Logic Controller, referred to as PLC, the tight coupling available with the Media Pro® 4000 is extremely beneficial. As a result, the Media Pro® 4000 System can do what it does best and allow other devices (such as PLC's) to do what they do best.

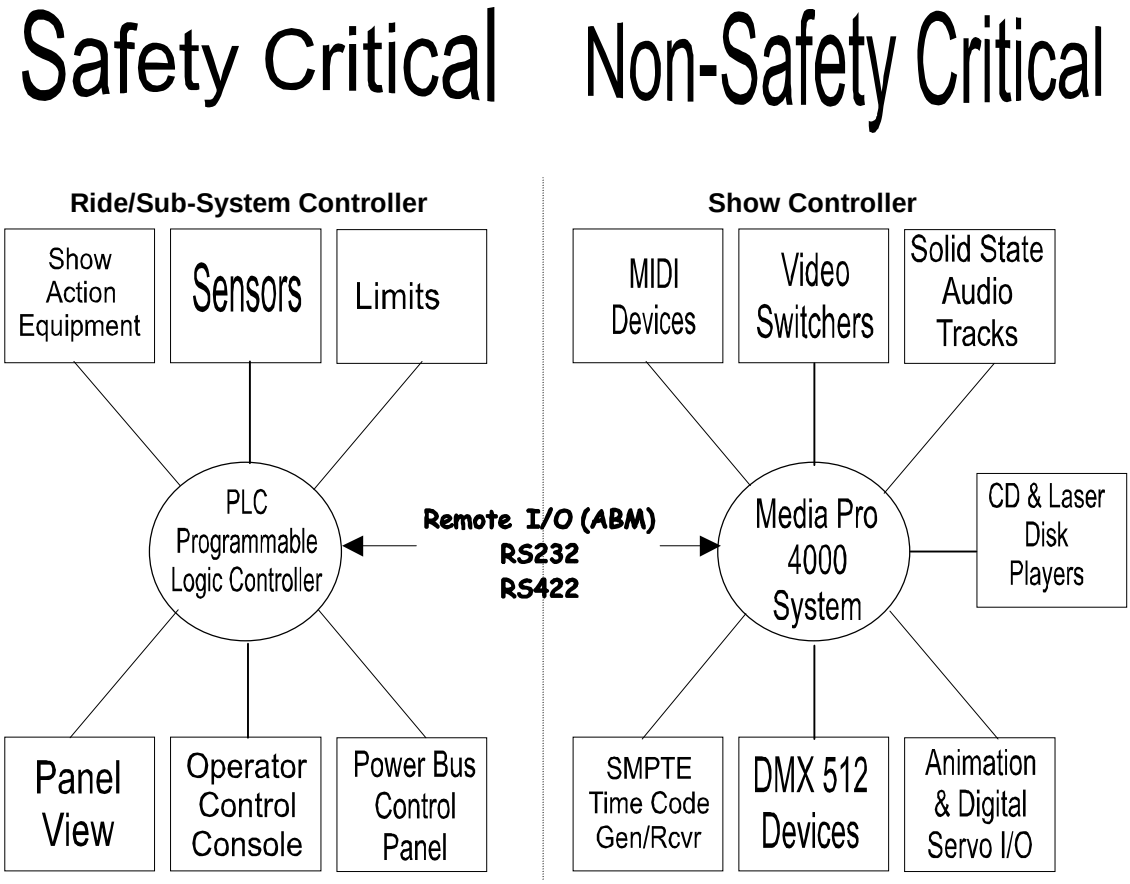
Synchronization to SMPTE, serial port control, deterministic real-time control are not what PLC's do very well, but these capabilities are the foundation of the Media Pro® 4000. To make use of the two systems where best suited, the Media Pro® 4000 contains two convenient ways to communicate with the PLC. Currently these implementations are for the Allen Bradley PLC5 and SLC500 family of programmable logic controllers. Future connections will be provided for other PLC types as the demand is warranted.

Direct connection via a DF1 protocol serial link embedded into the ICM and SEM firmware and can be used at any time. This RS232 connection can run up to 19200 baud and allow the Media Pro® 4000 to read and write memory locations in the PLC Data Tables. The DF1 protocol works only with rack 0. Since there are two serial ports on the ICM, two PLC's can be communicated with from one ICM. The Allen Bradley DF1 protocol allows up to 32 words of data to be exchanged in either direction providing commands to be sent to the Media Pro® or for the Media Pro® to send commands or data to the PLC. In most cases both are used and can provide a very powerful and economical solution over direct digital connect implementations. Diagram 1 (at the end of this section) illustrates a typical example.

An optional approach is to use an ABM 4010 module. This module plugs into the Rack 0 Media Pro® 4000 Module Rack and provides a direct connection to the Allen Bradley Remote Input Output (RIO) network. This makes the Media Pro® 4000 look like a remote rack to the PLC Processor. In addition, block transfers are also supported to increase the amount of data that a single PLC rack provides. This approach provides faster, isolated, and greater-distance communication between the two systems.

The ABM-4010 approach is especially powerful for multiple-vendor designs. This approach allows independent vendors to have the Media Pro® 4000 integrated into their sub-systems, each containing an ABM 4010 module. These can be single or multiple rack Media Pro® 4000 solutions each programmed by the respective supplier. The systems can be checked for acceptance by simply connecting a master PLC and demonstrating the required functionality. Once the systems are installed in the venue, all that is required is to connect the Remote I/O (RIO) from the master PLC to the sub-system ABM module and the sub-system is online and ready to be used by the master PLC. Defined show commands, data and fault information can now be communicated as required for the application. See Diagram 1 (following) for an example of this solution.

Diagram 1
Media Pro® 40000 – PLC Integration



Examples of the different devices that may be used in a show:

- MIDI
- Video Switchers
- Solid State Audio Tracks
- Digital Audio
- SMPTE Time Code
- DMX 512 Devices
- Animation & Digital Servo I/O
- CD & Laser Disk Players
  - Pioneer Laser Disk
- Show Action Equipment
- Sensors
- Limits
- Panel View
- Operator Control Console
- Power Bus Control Panel

1.6 Media Pro®4000 System Configuration

Parameters and Specifications –

The following list gives a concise overview of the system parameters and specifications:

- **RACK (CHASSIS/CARD CAGE)**
  - ☑ Every rack *MUST* contain an ICM (Intelligent Control Module). Located in the "p" slot, referred to as slot 17.
  - ☑ Power distribution on backplane – COM, CONSTANT, E/STOP bussed to every slot.

- **SMALL – Intelligent Module Controller**
  - ☑ IMC – either 1 slot or no slot, not expandable
  - ☑ Stand Alone, cannot be networked on MP Net
  - ☑ Used in kiosks, 'under-the-rock' applications.

- **MEDIUM – Horizontal Module Rack**
  - ☑ HMR – 5 slots, (ICM and 4 modules)
  - ☑ Can be networked, 1M bit

- **LARGE – Vertical Module Rack**
  - ☑ VMR – 18 slots, (ICM and 17 modules)
  - ☑ Can be networked, 1M bit

NETWORK

- ☑ UP TO 32 RACKS PER NET
- ☑ The master rack is RACK 0

MPCL - Media Pro® Control Language

- ☑ 512 CUES per show file
- ☑ 512 Events per Cue
- ☑ 1024 lines per cue
- ☑ Maximum of 20K character per Cue
- ☑ 512 Variables per Cue
- ☑ 512 Animation files per Cue
- ☑ Up to 2K Aliases per Show
- ☑ 1024-byte input block
- ☑ 1024-byte output block

1.7 Media Pro® 4000 System Components

This section describes the components of the Media Pro® 4000 system. Each module is summarized as to its contents and function along with any applicable software required for its use. *Please refer to the Media Pro® 4000 Brochure and module specification sheets for detailed information.*

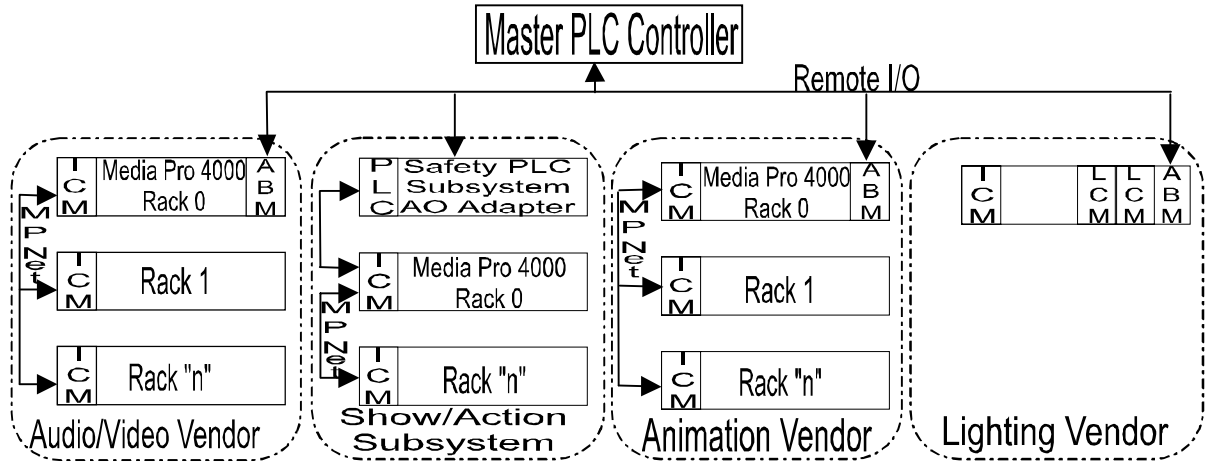
ICM-4020 – Intelligent Controller Module

The ICM is the System Master/Slave Intelligent Controller Module for Event and Real-Time Control Programming. The ICM is the nucleus module for the Media Pro® 4000 System and is required in each rack. The Master ICM is always located in Rack 0 and contains the show data. Each ICM in the system is configured based on the devices and resources connected to it. As the elements of a show are completed, they are downloaded to Flash Memory on the ICM using the MP4000 software. The features of the ICM-4020 are:

- 2 user RS232 ports
- 1 Maintenance parallel port
- 1 Maintenance RS232 Port
- V SYSNC In
- 1 M Flash Memory
  - contains the code, configuration, cue and animation information
- 1 Mbit Network

The ICM provides a built in Network called MP Net. This transformer isolated 1 Megabit network provides the inter-rack communication used by the Media Pro® 4000 system. The transformer isolation is imperative when racks are distributed across wide areas and numerous power sources. Using the built-in Maintenance RS232 port, a Hand-held Maintenance Terminal (HMT-4000) or laptop computer can communicate to any module in any rack in the system from any ICM on the network.

Diagram 2
Media Pro® 4000 Multi-Vendor Integration



ABM-4010 – Allen Bradley Module

The Allen Bradley Module provides AB **R**emote **I**nput/**O**utput (RIO) Network expansion for the Media Pro® 4000 system. It enables the user to directly communicate information to and from a PLC data table using Rack I/O and/or Block Transfers. The MP4000 Software supports this module and can be configured from ¼ to full rack of I/O on the RIO network.

AOM-4010 – Analog Output Module

The Analog Output Module provides sixteen (16) Channels of analog voltage output. Each channel may be configured for 8 or 12 bit operation. The output of each channel may be selected for 0 to 10 VDC or +/- 10 VDC output operation. There are no quantity restrictions for the AOM 4020. The MP4000 Software supports this module.

APC-4020 – Animation Programming Console

The Animation Programming Unit provides (8) digital and (8) analog inputs via a real-time animation programming console connected to the Media Pro® network. This allows an animator access to all racks connected to the network for programming animation scenes that include audio, video and mechanical effects. The unit requires a Pentium class laptop or tabletop computer running Windows 3.11 or Windows 95/98. It provides graphical display of digital and analog animation data. In addition, extensive editing capabilities are included in the software. The MPAPU4 Software Utility supports this unit.

ASM-4020/30 – Analog Servo Module

The ASM 4020 Analog Servo Module provides eight (8) Channels of non-compliant closed loop servo control or four (4) Channels of compliant closed loop servo control expansion for the Media Pro® 4000. Each channel may be configured for 8 or 12 bit operation. The module accepts a +/- 10VDC force feedback signal to implement the force compliant algorithm. The module can be downloaded with performance parameters for each channel's servo personality. Once the servo channels are tuned, each channel's device can be moved to a position from and event in a cue or data in an animation file. There are no quantity restrictions for the ASM 4020. The MP4000 Software supports this module.

DSM-4010 – Digital Sound Module

The Digital Sound Module provides Digital Sound expansion for the Media Pro® 4000 system. The module can be downloaded with audio data recorded on a PC using many available audio recording utilities. In addition, PCM, ADPCM and uLaw encoding/decoding schemes are supported. Once downloaded, these sounds can be triggered from cues or real-time animation as required by the application. There are no quantity restrictions for the DSM 4010. The MP4000 Software supports this module.

DSM-4020 – Long Play Digital Sound Module

The Digital Sound Module provides Long Play Digital Sound expansion for the Media Pro® 4000 system. A PCMCIA flash or ATA card can be inserted providing the memory storage for long play applications. Modules can be downloaded with audio data recorded on a PC using many available audio recording utilities. In addition, PCM, ADPCM and uLaw decoding schemes are supported. These sounds can be triggered from cues or real-time animation as required by the application. There are no quantity restrictions for the DSM 4020. The MP4000 Software supports this module.

HMR-4000 – Horizontal Module Rack

The Horizontal Module Rack provides an external 115/240 VAC Power Supply and motherboard for the Media Pro® 4000 Modules in a 5.25" vertical 19" Retma Rack package. This unit has the capacity to hold one (1) Intelligent Controller Module (ICM-4010) and up to four (4) Media Pro® 4000 Series interface/control modules. In addition, two (2) field power distribution busses are provided on the backplane for Constant and Emergency Stop connections.

HMT-4010 – Hand-held Maintenance Terminal

The HMT 4010 is a hand-held ASCII Terminal with a 4 line by 20 character LCD Display. This unit plugs into the ICM Maintenance serial port to provide system control and status information. User Cues can send messages to this display for system operations.

IMC-4010 – Integrated Module Controller

This unit is usually configured in kiosks or very dedicated uses. It does not support the Media Pro® Net and cannot be networked. The IMC 4010 provides 3 serial ports that may be configured by the user along with a maintenance port. Port 4 supports DMX-512 controlled devices. These ports are typically used to control devices such as Laser Disk players, CD Players, Video Switchers, Lighting Control Boards and Programmable Logic Controllers. The IMC also has an 8-bit input and an 8-bit relay closure output, which are configured as an IOM-4020 on slot 0. Each port is controlled by the Media Pro® 4000 software using the same MPCL language.

IMC-4020 – Integrated Module Controller

The IMC 4020 provides all the same features of the IMC 4010 with one expansion slot. It also has a rack mount enclosure.

IOM-4020 – Input / Output Module

The Input / Output Module provides 24 VDC optically isolated digital input and output expansion for the Media Pro® 4000 system. Modules can be connected to external switches and indicators providing a direct connection to these physical devices. The 1 amp outputs may be configured to deliver animation data or controlled to the bit level using event commands in a Cue. There are no quantity restrictions for the IOM 4020. The MP4000 Software supports this module.

LCM-4020 – Lighting Control Module

The lighting Control Module provides 5 optically isolated DMX 512 outputs for the Media Pro® 4000 system. Each of the 32 logical ports can be individually programmed using the cue events, timed looks or real-time DMX recording techniques. These resources are stored on the module using a PCMCIA ATA FLASH Disk memory device. Once the board is loaded with the show resources they can be controlled by the system ICM and played back in sync with the other resources of the application. There are no quantity restrictions for the LCM 4020. The MP4000 Software supports this module. Parameters include:

- 512 Channel DMX
- Single Fader (1 Cross Fade – up or down)
- Program wait/, up/, down/ in tenths of a second
- 32 Addressable ports
- All Run independently, pile-on to single ‘stage look’

SEM-4020 – Serial Expansion Module

The Serial Expansion Module provides eight (8) RS232 serial ports for the Media Pro® 4000. Drivers in the system firmware directly support many of the devices common in our industry. There are no quantity restrictions for the SEM 4020. The protocol for each port is defined during system configuration and is downloaded using MP4000 Software. The MP4000 Software supports this module.

TCM-4020 – Time Code Module

The Time Code Module provides SMPTE Time Code generation and synchronization capabilities to the Media Pro® 4000 system. Any number of modules can be added to provide synchronization of numerous cues and real time animation data to multiple sources of SMPTE. In addition, external video can be connected to the TCM video sync input to provide SMPTE phase locked to external video devices. There are no quantity restrictions for the TCM 4010. The MP4000 Software supports this module.

VMR-4000 – Vertical Module Rack

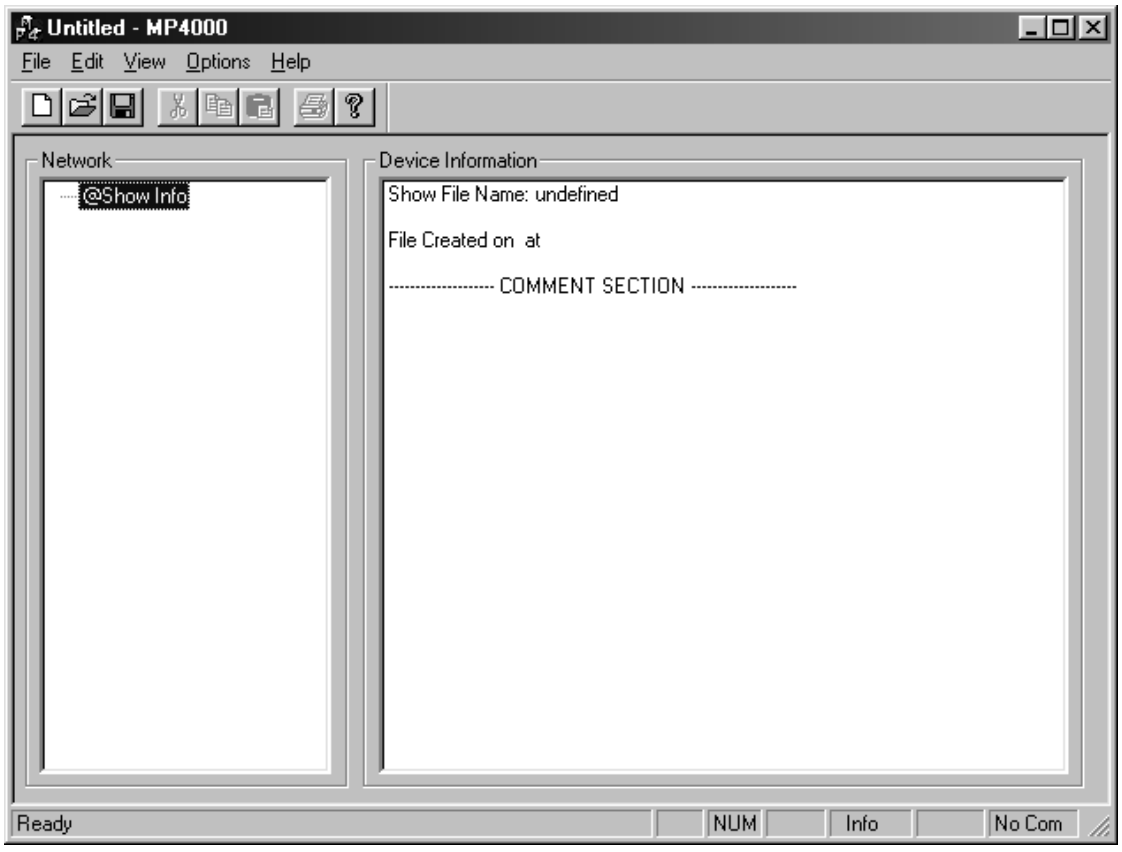
The Vertical Module Rack provides a 115/240 VAC Power Supply and motherboard for Media Pro® 4000 Modules in a 10.5" vertical 19" Retma Rack package. This unit has the capacity to hold one (1) Intelligent Controller Module (ICM-4010) and up to seventeen (17) Media Pro® 4000 Series interface/control modules. In addition, two (2) filed power busses are provided on the backplane for constant and Emergency Stop connection.

2 Section 2 – Software Operation

This section details how to use the Media Pro® Software and how to configure the system. The programming of the cues and command specific reference is in section 3.

2.1 Starting the Media Pro® Software

From the Windows desktop, highlight the **MP4000** icon and double-click to open the program. The following screen appears:



The buttons and icons on the screen are standard Windows 95/98 functions.
At the top left of the screen is the name of the current file in use, in this case it is Untitled – MP4000.

The **Network** pane (box on the left) displays the physical configuration of the network.
The **Device Information** pane displays the current show file being processed.

The status line at the bottom of the screen displays messages relative to the file in process.

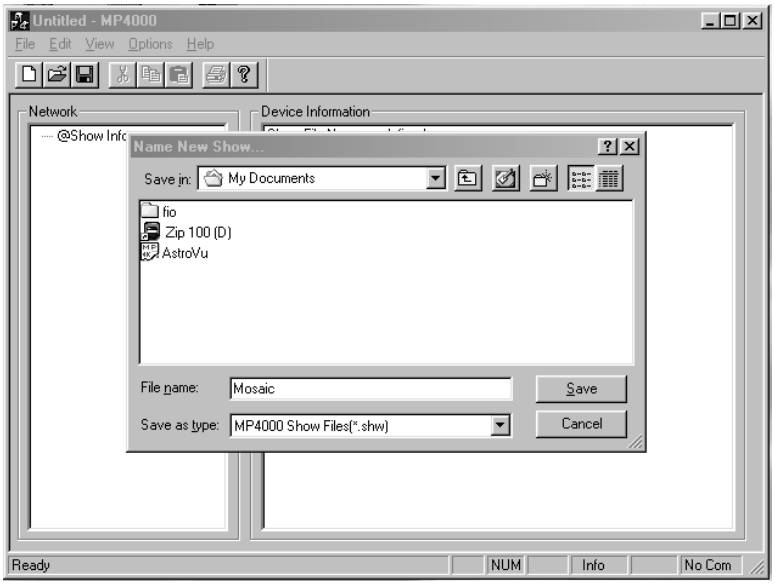
2.2 Configuring the System Modules

Before the system software is configured, it is recommended that the programmer and/or system designer write down the physical layout. You will need to know what devices will be attached to which modules, the names of the aliases and routines to be used, etc. Once that has all been established it is time to configure the software.

The ICM is the controller of the system and is the first module to be configured. This section will show how to set-up the ICM configuration in the Media Pro® 4000 software. For the other modules, please refer to that module's User Manual.

2.2.1 Opening Show File

Show files contain all information relating to a show. It includes the system configuration and the programming (cues). The first step in defining a show is to configure the physical layout.



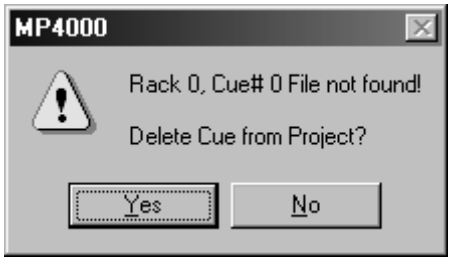
There are two ways to open a show file: define a new show or open an existing show.

New Show –
Select New from the File pull-down menu. A message window appears and prompts for the show file name:

For this example, the show name is **Mosaic**. Type in the filename then click **Save**.

All show files are given the extension .**shw**.

NOTE: Show files must *not* have Read only properties. If *any* of the show files are **READ ONLY** then error messages appear when trying to open an existing show file.



All show files must have read/write access in order to use them with the MP4000 software.

Exit the MP4000 software and make all show files accessible, by either changing the file properties or by placing them on a drive that has read/write capabilities. Refer to Microsoft Windows Manual for details about file attributes.

The title bar displays the show filename.

The **Network** pane displays **Rack 0** information

the **Device Information** pane now displays the details about the file ***Mosaic***

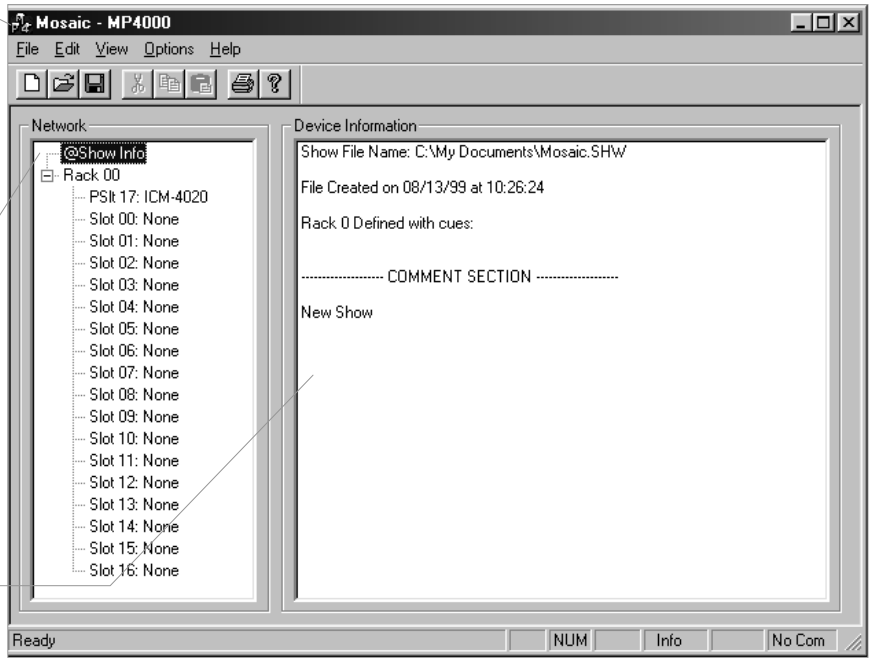
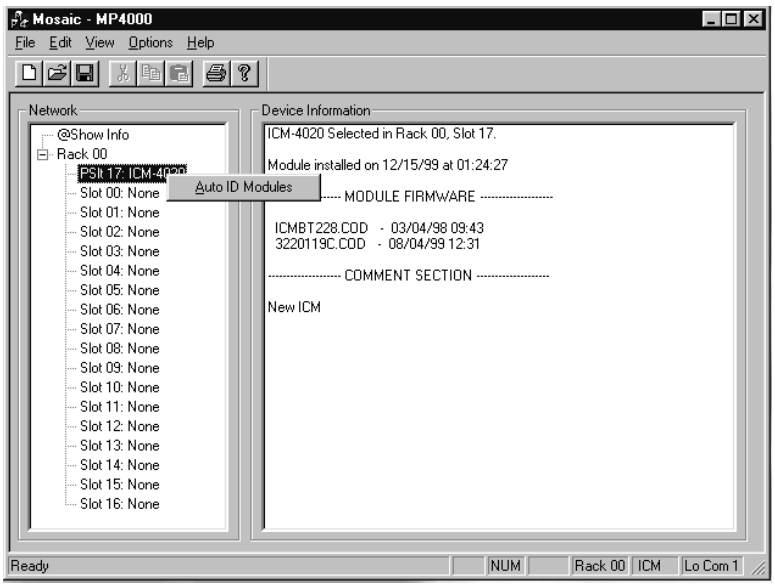


Figure 2 – 1**. MP 4000 System Menu.**

The next step is to configure the system.

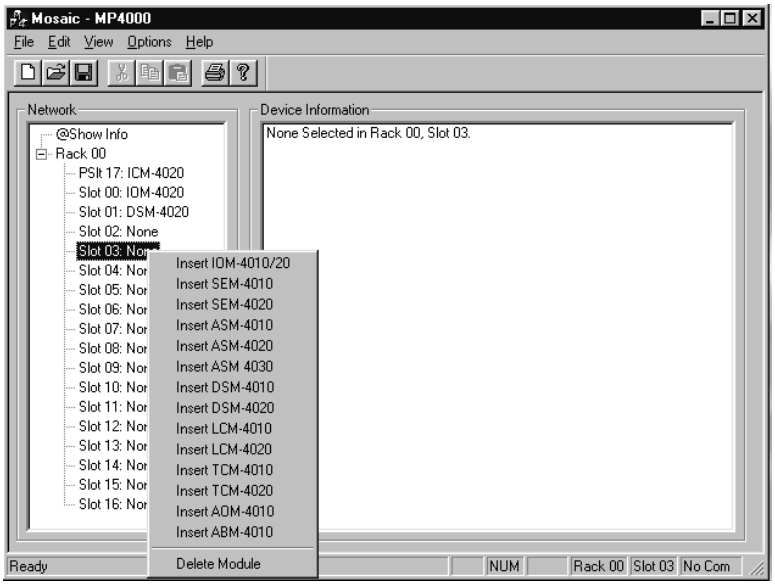
2.2.2 ICM System Configuration

There are two methods of configuring the system, automatically or manually.



Manually –

To manually configure the system (when the rack is not connected or available), left click on the slot to configure and right click. The device menu appears.



A list of modules is displayed. Highlight and right click on the desired module.

When a device is added and the software does not find it, a message will display in the Device Information pane:

Module Not Present or Wrong Type

Firmware not Available

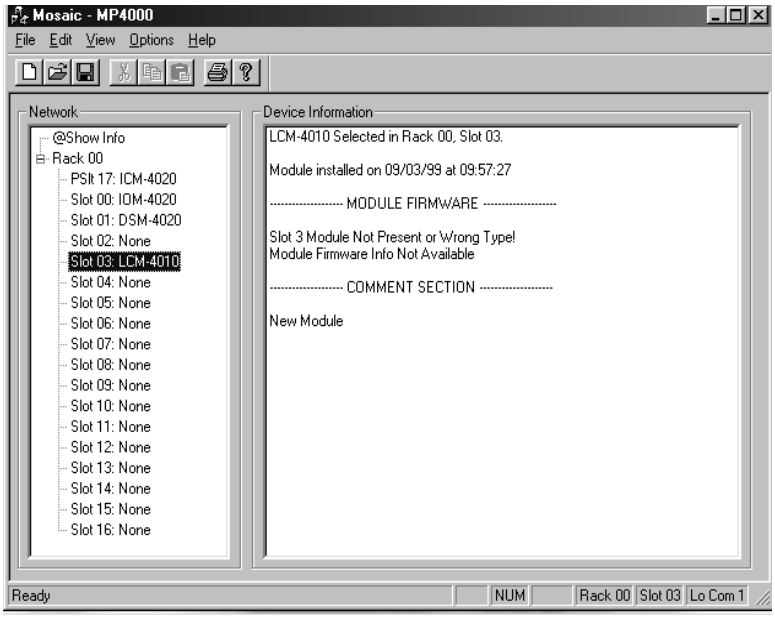
It may be added and configured even if it is not found. Then the information can be downloaded when the module is physically installed.

Automatically –

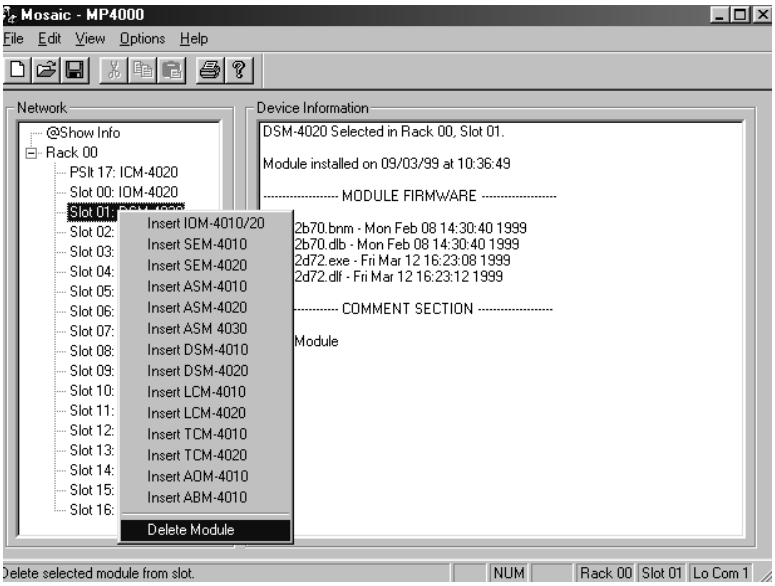
If the rack you want to configure is available (installed and connected correctly), then the MP4000 software will automatically configure the system.

Left click on the P SLOT of the rack to highlight the selection.

Right click on the P SLOT, and a drop down menu appears. Left click on menu and the configuration information will be shown in the Network Pane.

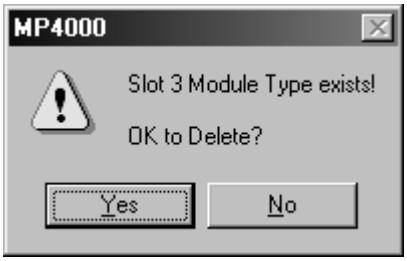


If a module needs to be removed from the rack configuration, highlight and select the appropriate slot, then right click to bring the device selection list on screen. At the bottom, highlight and right click on



Delete Module –

A warning message appears:

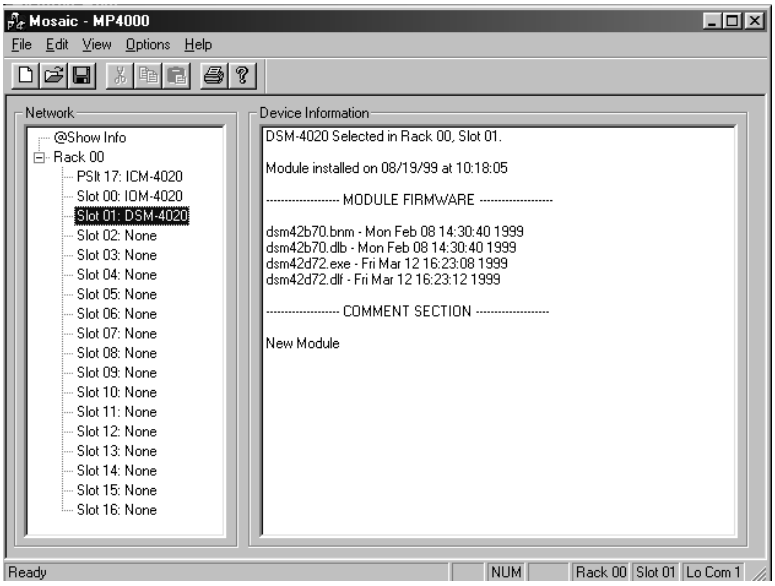


Click on the Yes button. The Module is removed from the list of devices in the rack.

After the modules have been configured, the information about the module can be displayed in the Device Information pane.

Select and highlight the module to review and right click the following type of information will be displayed:

- The device model, rack, and slot identification.
- The date of installation.
- The firmware revisions.



2.2.3 Configuring Multiple Racks

In systems that have more than one rack for their purpose, each rack must be configured in the ICM software. Before the software will recognize the racks, three steps must be taken:

- Connect the racks together
  - There can be a maximum of 32 racks
  - Refer to the appropriate manual for pin out and connection information.

- Address the racks correctly
  - Each rack has a blue switch on the backplane that selects from 0 to 15.
  - For addressing from 16 to 31 a jumper is placed on the backplane which allows the upper address selection.

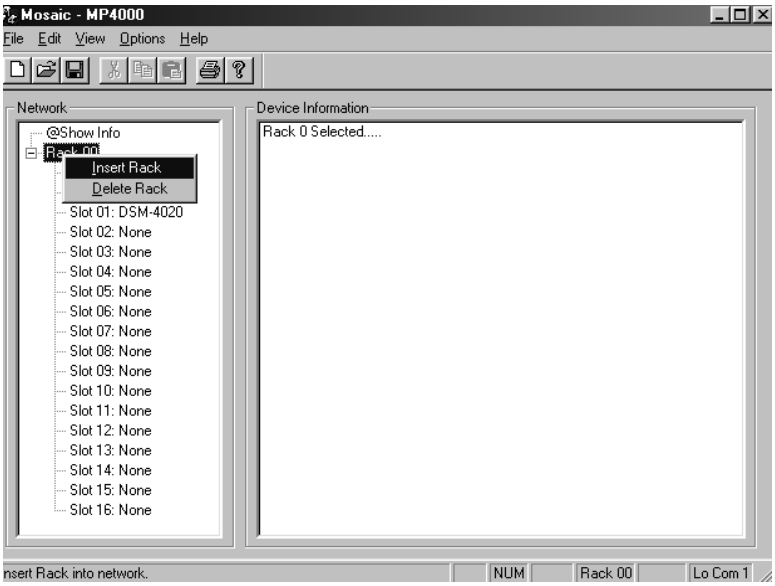
- Configure the MP4000 software
  - Insert the racks in the ICM menus.

Adding Racks –

In systems that have more than one rack, the racks are added by highlighting Rack 0 and left click to select it. Then right click to bring the up the sub-menu.

Select **Insert Rack**
then right click.

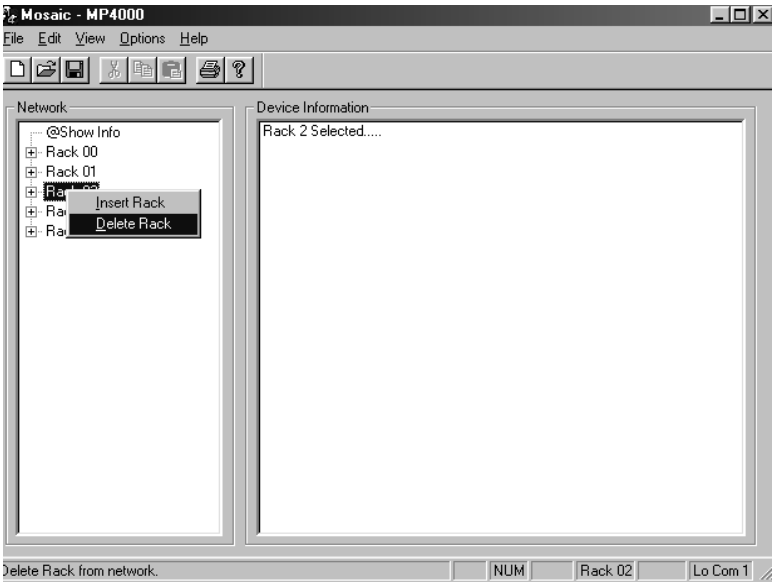
Note: It will only add a rack after the last rack on the list.



Deleting Racks -

To delete a rack, highlight and left click on the rack to remove and right click. The sub-menu appears, highlight and left click on **Delete Rack.**

The rack is removed from the list.

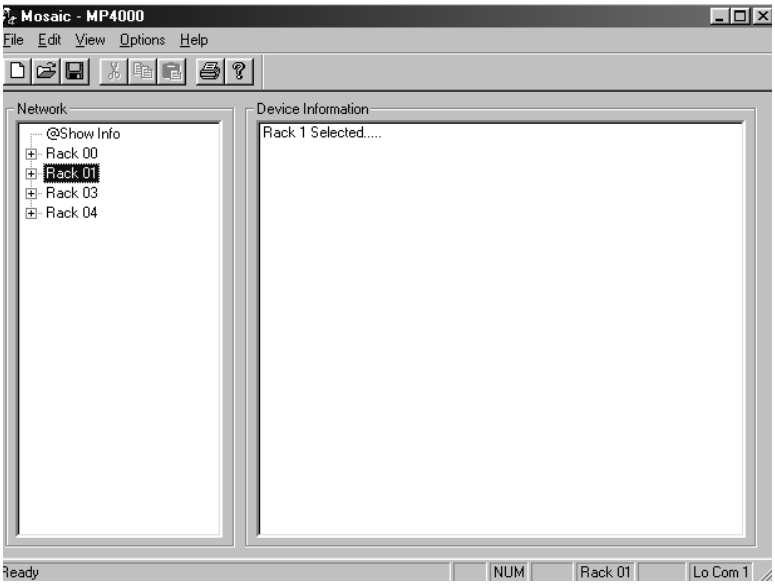


After the rack has been deleted, the information in the screen will show the remaining racks.

It does not renumber the racks. This is due to the physical addressing on the backplane that is required.

Note: It will not allow RACK 0 to be deleted.

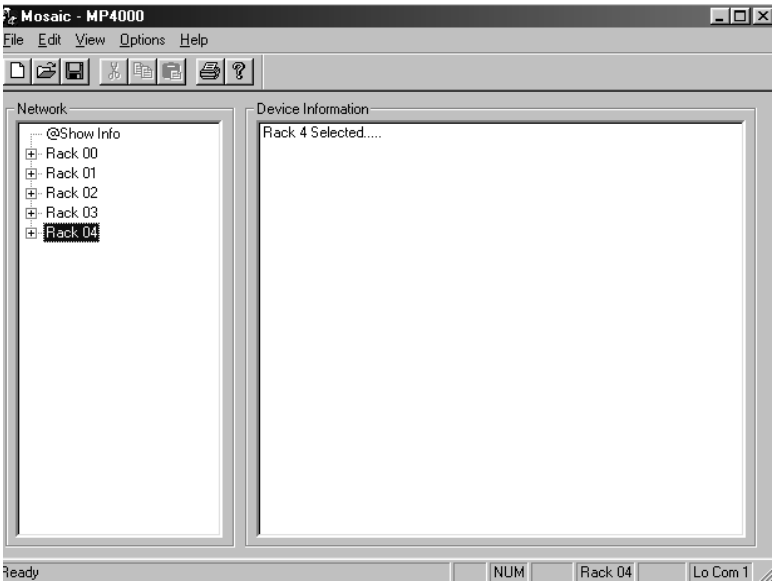
Once the devices and racks are configured, the next step is to define the system parameters.



Expanded Display –

NOTE: When looking at the system configuration in the network pane, it will show all the slots by default.

If you want to condense the information, position the cursor next to the rack minus sign, left click and it will show only the racks for the system without the individual slots.



2.3 Cue Editing

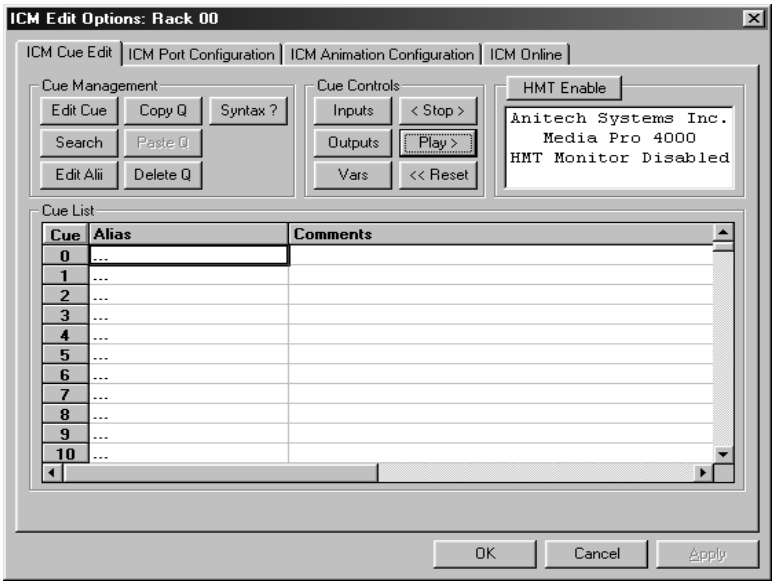
To start defining the show parameters left click twice on the ICM in Rack 0. The ICM edit menu appears. There are four option tabs on this menu:

- ICM Cue Edit
- ICM Port Configuration
- ICM Animation Configuration
- ICM Online

2.3.1 Cue Edit Menu

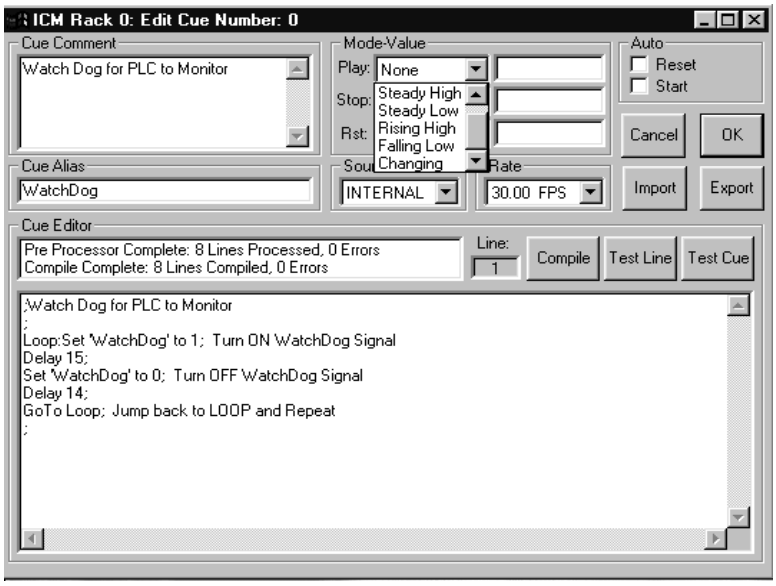
Define the Cues from this menu. The buttons in the Cue Management pane allow basic editing functions of the cue that is highlighted in the Cue List pane.

To edit a cue, position the cursor and highlight the appropriate cue. Next, either double left-click the mouse or press the Edit Cue button and the Cue Edit window will be displayed.



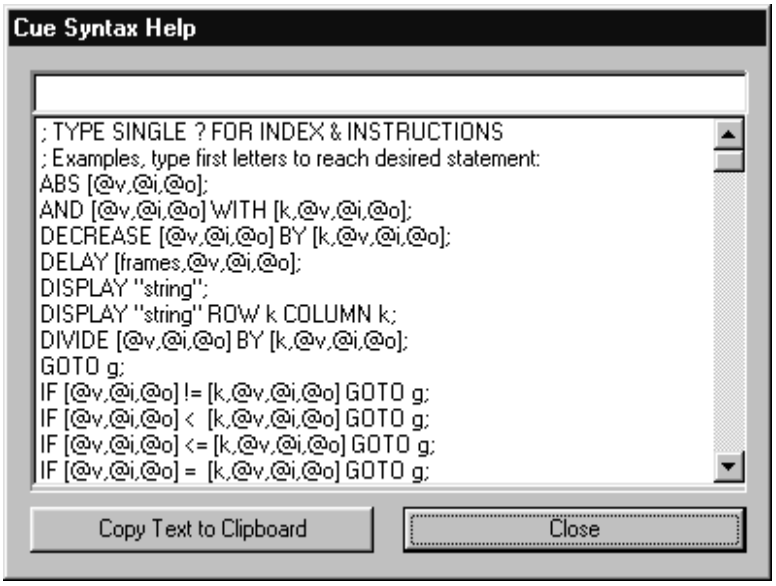
Enter the Cue information –

- ☑ Place a comment in the comment section to describe what is happening in the cue.
- ☑ If desired, give the cue an Alias to be used in the PLC, up to **32** characters. Use underscores for spaces.
- ☑ Type in the Media Pro® Control Language commands for the cue in the Cue Editor pane. Refer to the programming section for more details.



Syntax

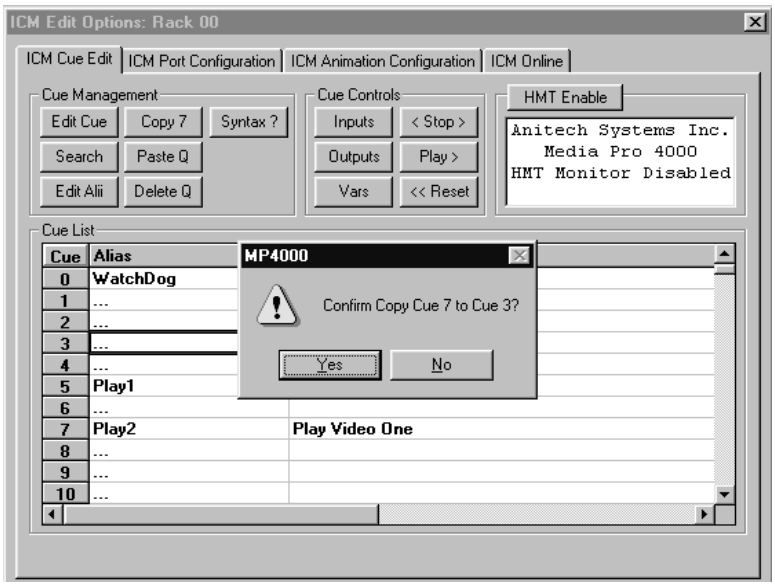
When entering the cue information, the Syntax ? button will display the Cue Syntax Help screen.



For convenience, the commands may be highlighted and copied to the cue and then edited for specific parameters and values.

Copy and Paste Cues –

For cues that are similar and repetitive, they may be written once and then copied. To copy a cue, position the highlight over the cue to be copied. Click on the Copy Q button on the Cue Management pane. It will change to Copy N (where N will be the cue number to be copied).



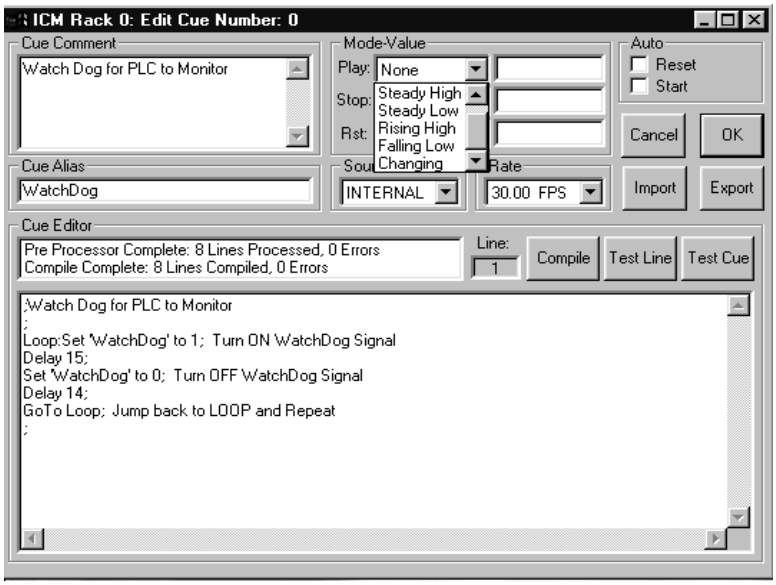
Then place the highlight over the cue number to paste the cue into that cue number. Then click the Paste Q button in the Cue Management pane.

A message window appears confirming the paste. Press the Yes button to perform the paste.

Cue Mode-Value –

Cues may be triggered by a variety of methods. The cue may be set to Play, Stop, and Reset by the following trigger values:

- None
- Steady High
- Steady Low
- Rising High
- Falling Low
- Changing



Highlight the appropriate pane in the Mode-Value section, click on the down arrow and a pull-down menu appears. Select and click on the desired value.

Each cue may be designated to be Reset or Start automatically. To select these automatic functions, place a check mark next to the desired selection in the Auto pane at the top right of the Cue Edit screen.

There are two other settings that can be selected on this menu – the Source and the Rate:

- SOURCE - The default source is the internal clock. The other option is SMPTE. Click on the scroll bar to bring out the menu, highlight SMPTE and click again.
- RATE - The Rate specifies the Internal clock rate at which the cue will be clocked. The choices are:
  - 24FPS
  - 25FPS
  - 29.97FPS
  - 30.00FPS

Compiling Cues –

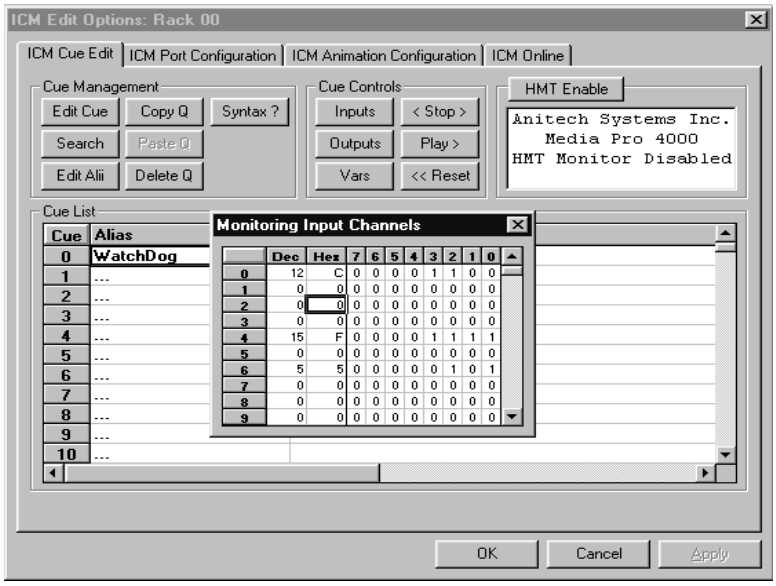
After entering the cue information, click on the Compile button and the Media Pro® software will check the cue for errors. The messages are displayed in the Cue Editor pane. To the right of the pane is a Line pane that will display the line number of the cue that is being edited.

The cue may be tested by line and also in its entirety. To test the line, place the cursor in the line to test and click on the Test Line button.

To test the entire cue, click on the Test Cue button. The cue is sent to the ICM and tested.

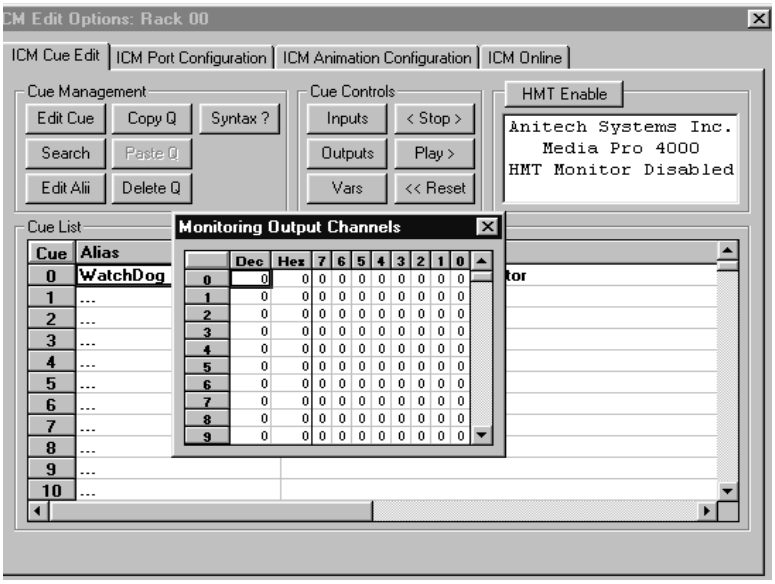
2.3.2 Cue Controls

Input Channels –



- ✓ There are 1024 Input Channels for each show. They are used to communicate with the various modules of the system. The information is displayed from 0 – 1023.
- ✓ Click on the Inputs button in the Cue Controls pane.
- ✓ Information may be entered by Decimal, Hexadecimal, or Binary formats. Move the slider bar until the channel to be set is shown in the Monitoring Input Channels pane. Type the correct information and press enter.

Output Channels –



There are 1024 Output Channels for each show. The information is entered the same as for the input channels. Click on the Outputs button in the Cue Controls pane. The Monitoring Output Channels pane is displayed.

To close either of the Channels pane, click on the **X** in the upper right corner of the title bar.

Note: Animation files will always reside in the first 512 Output Channels from 0 - 511. It is recommended that the other modules communicate using the remaining channels from 512 – 1023.

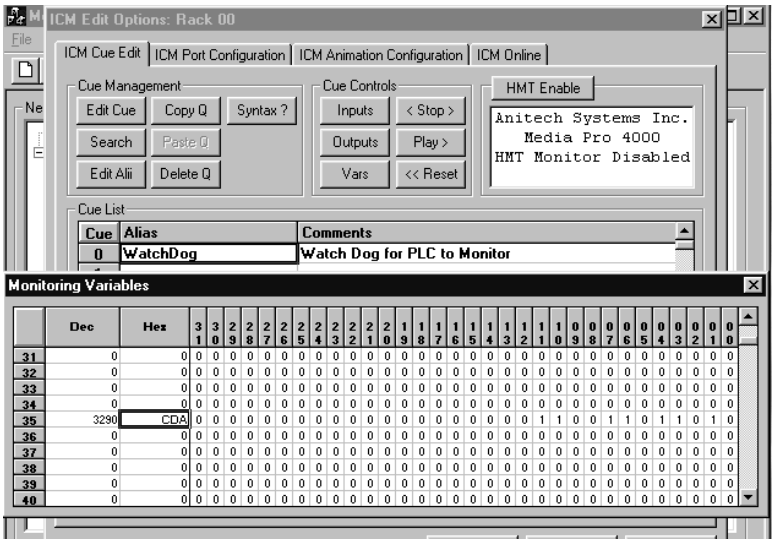
For modules that need 12 bits of information on the channels, use even-numbered channels for the first 8 bits and the consecutive odd-numbered channel for the remaining 4 bits.

Configuring Variables –

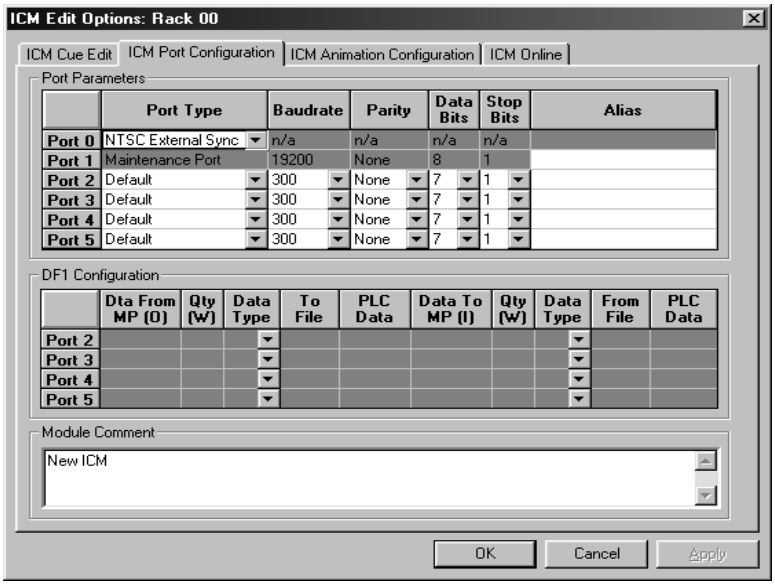
There are 512 variables available for use when programming the modules. They are numbered 0 – 511. The information is represented in Decimal, Hexadecimal and binary formats.

To configure variables, click on the Vars button in the Cue Controls pane.

Position the slider bar until the variable to be set is shown in the window and enter the desired value. Press the **X** on the right side of the title bar to close the window.



2.3.3 ICM Port Configuration



The second tab on the ICM edit options menus is the Port Configuration menu.

Port 0 –
Click on the scroll arrow to view the selections for the port configuration. Highlight the desired port type and left click.

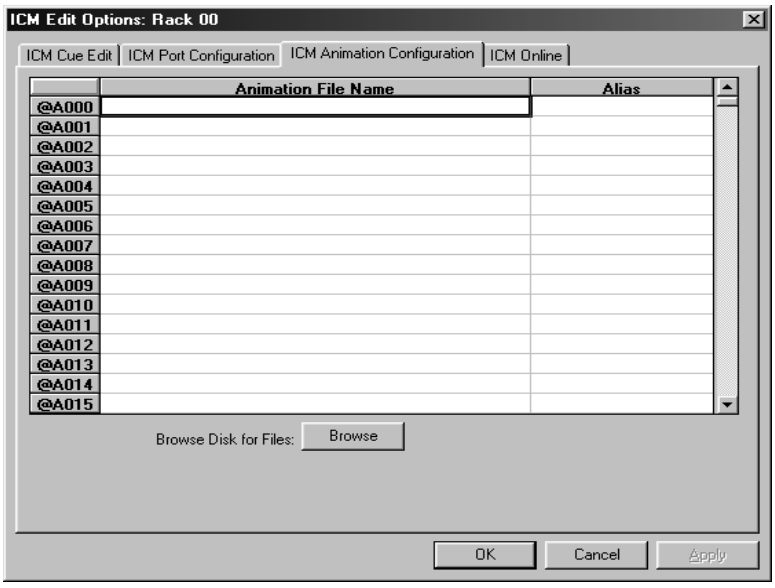
Port 1 –
Reserved for maintenance.

Ports 2 through 4 –

Click on the scroll arrow to view the protocols for the port configuration. Highlight the desired port type and left click.

Port 5 is reserved for future development.

2.3.4 ICM Animation Configuration



Enter the name of the animation file in the highlighted line.

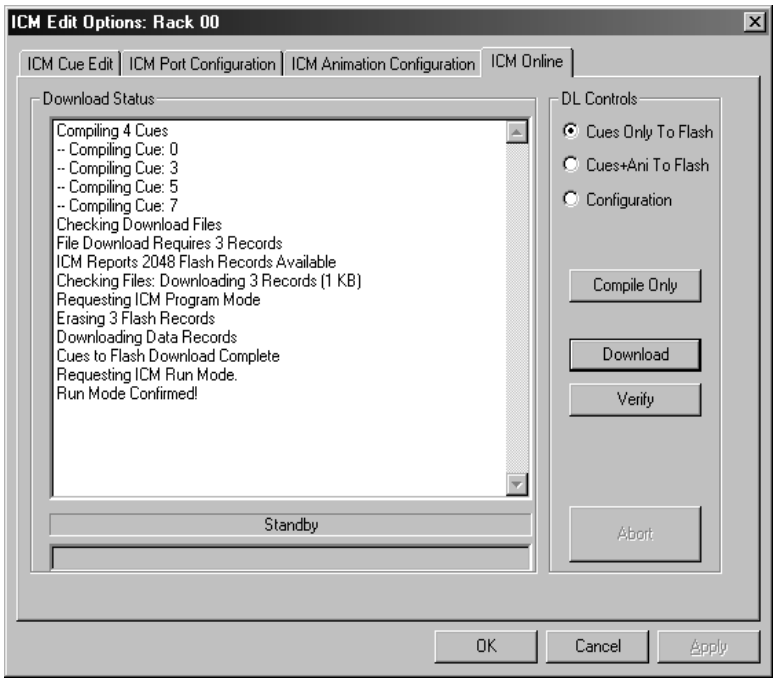
The BROWSE button brings a pop-up menu to select and highlight a path / filename for the animation file. Either double-click on the filename, or press the OPEN button and the filename is placed on the animation list.

Type in the Alias, as appropriate.

2.4 ICM Online

The system needs the cue, configuration, and animation downloaded in order to operate. Every time a change is made to any of these items, they need to be downloaded again. If changes made to a cue do not take affect, try to download the information and then re-run the cue.

To send the information to the ICM, select the ICM Online Menu.
The DL Controls area of the screen has three choices:



✓ **Cues Only to Flash –**
Downloads *only* the Cue files to Flash.

✓ **Cue+Ani To Flash –**
Downloads the Cue *and* the Animation file to the Flash.

NOTE: The entire Flash is re-written each time a download is run. Therefore, the previous animation file will be wiped out with a Cues Only download.

✓ **Configuration –**
Downloads the Configuration information. No Cues or Animation files are affected. Whenever there is a change to any configuration of the modules, it is necessary to send the configuration information to the ICM. When done, the message "Show Config File Build Complete".

Compile Only –
Before downloading the information to the ICM, the files need to be compiled. Select COMPILE ONLY button. If there are any errors, it will be displayed in the Download Status window. If no errors are found, the message "Cue File Complete" displays.

Download –
Select the correct option by clicking the circle next to it, a dot will be displayed. When ready to download, select the Download button. The information selected to be downloaded will be sent to the ICM. It checks the Flash for record size availability, and states the record size being downloaded. A record is 512 bytes. When completed, it will give a message "Cues to Flash Download Complete".
NOTE: The ICM must be switched into Program Mode on the front panel in order to download the files.

Verify –
It is strongly recommended to perform a verify after a download. This ensures the download has completed successfully.

2.5 System Configuration Reports

Once the Media Pro® system has been programmed and set up, a report of the configurations can be printed for reference and future revisions.

To print a report, select PRINT from the FILE options at the top of the menu screen. Place a check mark next to the reports to print. For the Rack and Cue reports type in the number(s) of the reports to print, ie: to print cues 0 through 21 enter the start and end numbers in the boxes next to the Cue report selection.

There are six different reports that may be generated:

System Summary –
Prints the Show Configuration Summary. It contains the name of the show and the racks in use. It will print all available racks even those not in use.

Alias Table Report –
Prints the Show Alias Summary. It contains all the Alias strings used in the entire show. They are sorted in ascending order by String and by Object.

Channel Report –
Prints the Show Channel Report. It contains the Channels that have been assigned for the entire show. The Output channels are printed first and the Input afterward.

Animation Report –
Prints a listing of the Animation files used in the show. It contains the Animation file names and their path\filename locations.

Configuration Report –
Prints the Rack Configuration Report. It contains a listing of each rack / slot assignments and the configuration of the modules. It specifies the port configuration, channel assignment, and alias.

Cues Report –
Prints a report of the Cues used in the show. It contains the Cue Number, Alias, Cue Header, and Cue Text. This report can be very lengthy, so when any changes have been made to cues, just enter the number of those cues changed in the start and end boxes. It will only print the cues that have information in them. It will not print empty or unused cues.

3 Section 3 – Media Pro® Control Language

This section details how to program cues for the Media Pro® system. First, it defines terms and concepts for programming applications. Second, it describes the commands (verbs) used in the Media Pro® Control Language in detail.

3.1 MPCL Introduction

The Media Pro® 4000 is programmed using the Media Pro® Control Language or MPCL. This language was developed by Anitech Systems and is loosely based on the Media Control Interface or MCI specification created by Microsoft. It is designed to allow numerous devices to be controlled by a common set of verbs and tokens. This allows a designer to learn one syntax no matter what type of object is being controlled.

Modules in the Media Pro® Family can contain up to thirty-two (32) addressable ports. For instance, an IOM has two 8 bit digital ports referenced as Port 0 and Port 1. The DSM has two audio ports referenced as Port 0 for the Left Port and Port 1 for the Right Port. In the case of a SEM, each of the 8 serial ports are directly addressable as Port 0 to Port 7. This allows multiple resources, or ports, to be identified on a module within a given Rack and Slot location.

The basic concept of MPCL is that an object, whether it be a serial device, solid state audio track, internal variable, or logical channel, it can be referenced in a common way. These references always contain a Type and an Address. A reference can be made either directly or indirectly by using an Alias Name.

The most commonly used commands:

If	Causes the Cue to jump forward or backwards in the program based on specified condition
Load	Tells a resource to Load
Play	Tells a resource to Play
Ramp	Causes a resource or device to ramp up or down to a specified level
Seek	Tells the resource to go to a track or segment
Set	Causes resources and devices to be placed at a specific value or condition
Stop	Tells a resource to stop
Unload	Tells a resource to unload

3.1.1 MPCL Terminology

ALIAS An English representation that references an OBJECT. It is contained within single quotes. Ex: 'PRESHOW_LDP'=@r0,3,4

ASCII A numeric code used by computers to represent characters entered from keyboards or transmitted in data communications. A space bar pressed on a keyboard is represented by an ASCII 32.

COMMENT Text following a semicolon on an EVENT line within a CUE.

CUE A list of up to 512 **EVENTS** containing verb, object, token and variable items.

EVENT A single line of verb, object, token and parameter items that are contained in a CUE.

LABEL A name, followed by a colon, used on a line of a CUE for a forward or backward jump.

OBJECT Term used to *reference* a resource within the Media Pro® 4000 system. Contains TYPE, RACK, SLOT, PORT. Identified with the **@** symbol. Ex: @r0,3,4 refers to rack 0, slot 3, port 4. @v25 references variable 25.

PARAMETER The item that follows a token. It can be a number, time code, ON, OFF etc. In certain circumstances an object may be passed as a **PARAMETER**.

PORT Addressing reference to a resource on a Media Pro® 4000 system. Modules can contain up to thirty-two (32) ports.

RACK A card cage containing Media Pro® 4000 Modules. Address range from 0 to 31.

RESOURCE A controllable device connected to or contained within the Media Pro® 4000 system. Ex: a Laser Disk connected to a serial port, or a DSM Sound Track on a DSM board.

SLOT A physical location in a **Rack** that contains a Media Pro® 4000 Module. They are addressed from 0 to 17. SLOT 17 is also known as the "P" SLOT, which is reserved for the **RACK** ICM.

STRING A collection of alphanumeric characters contained in double quotes. Used to add comments to the cue commands. Ex: "This is a string".

TOKEN Reserved words for modifiers of the verbs used in a **cue event.** The list of TOKENs used by the Media Pro® 4000 software are on page **22,** table 2 - 2**.**

TYPE The kind of **OBJECT** being referred to in a direct addressing statement. The following is a list of addressing **TYPE**s currently allowed in cues:

R	Rack, Slot, Port Address	@r31,16,31
I	Logical Input channel	@i1023.7
O	Logical Output channel	@o1023.7
V	Variable address	@v511.31
Q	Cue number address	@q511
A	Animation number address	@a511

There may be up to 2000 Aliases, 1024 lines, and 20,000 characters in a cue.

VARIABLE A 32-bit value that can contain a value (number) used in a cue line. There is a maximum of 512 variables in the ICM. Variables are referred to from @v000 through @v511. A variable may be assigned an **ALIAS**.

VERB Reserved words for the actions (commands) used in a **CUE EVENT**. The list of VERBs used in the Media Pro® program software are listed on page 22, table 2 - 1.

3.1.2 Direct Addressing

Direct addressing can reference any object in the system usable in cues. All direct addresses must begin with an **@** character. The next character defines what TYPE of object it is. The rest of the field defines the value of the address. The format is: ***@ Type Address***

Example:

PLAY **@R1,1,1;** Sends a play command to the device in Rack 1, Slot 1, Port 1.

PLAY **@A000;** Plays the Animation File stored in the ICM in @a000.

SET **@O600 to 25;** Sets the Logical Output Channel 600 to the value 25.

SET **@O602.3 to 1;** Sets the Logical Output Channel 602, Bit 3 to a 1 value.

The following is a list of addressing **TYPES** currently allowed in cues:

Table 3 - 1

SYNTAX	DESCRIPTION	EXAMPLE
R	Rack, Slot, Port address	@R31,16,31
I	Logical Input channel	@I1023.7
O	Logical Output channel	@O1023.7
V	Variable address	@V511.31
Q	Cue number address	@Q511
A	Animation number address	@A511

3.1.3 Alias Addressing

To simplify the references of these devices, an Alias can be assigned to the ports during the system configuration. Examples of this process are in Section 3.0. When the devices used in the previous example are defined, a fifteen-(15) character name can be assigned to them. To use an alias in an event it must be enclosed in single quotes. In this case, the form is **'ALIAS_NAME'** or as an example:

PLAY 'PRESHOW_LDP'; Sends a play command to the device in Rack 1, Slot 1, Port 1.

SET 'PLC_WATCHDOG' to 25; Sets the Logical Output Channel to the value 25.

PLAY 'SCENE1_ANI'; Plays the Animation File in the ICM in @a511.

3.1.4 Cues

The Master ICM in a system provides a set of up to 512 'Cues' which becomes the master script for the show presentation. The Cues are written using a PC-Windows-based software program called MP4000. Each Cue Sheet can contain up to 512 lines of Events.

Each Cue has a header that describes how the Cue operates, how it starts/stops, etc., and event lines specifying the activity of the cue. This is described further in the examples presented in Section 3.0.

Each Event line has the following format:

(optional)	*required*	*required*	*as needed - 32 bit number (object could be used as a token)*					
LABEL	**VERB**	**OBJECT**	**Token 1**	**Param 1**	**Token 2**	**Param 2**	**Token 3**	**Param 3**

Not all fields are required depending on the verb being programmed. Entering a semicolon denotes comments. Any text after a semicolon is treated as a comment and will not be sent to the ICM. All event lines must end with a semicolon. Following are additional syntax rules:

The editor ***IS*** case sensitive for Labels and Alias Names.

It will also allow embedded spaces or tabs.

Tabs in the cue editor can only be entered with a **Ctrl-Tab** combination.

There are two buttons - import text and export text, which will read from disk or save to disk. Files are ASCII text file format. The import will insert the file text at the cursor position. The export stores the highlighted text.

Constants can be defined in any of the parameter fields. There is enough storage for a 32-bit number (4 billion), depending on the Object Type the range could be smaller. Some events will take a Time Code as one of the parameters using the syntax HH:MM:SS.FF. This Time Code format is internally converted to frames based on the frame rate for the cue. The Parameter can also be entered in frames directly.

3.1.5 Verbs

Table of the current **VERB** set for the Media Pro® 4000 cues:

Table 3 – 2

ABS	LOAD	
AND	MULTIPLY	SEEK
DECREASE	NEGATE	SEND
DELAY	OR	SET
DISPLAY	PLAY	STEP
DIVIDE	RAMP	STOP
GOTO	RANDOM	UNLOAD
IF	RECORD	WAIT
INCREASE	RESET	XOR
INVERT	RESUME	

3.1.6 Tokens

Table of the current **TOKEN** set for the Media Pro® 4000 cues:

Table 3 – 3

!=	DISK	RAMP
<	DOWN_IN	RATE
<=	FADE_IN	REVERSE
=	FADE_OUT	ROW
>	FADER	SEGMENT
>=	FOR	SET
ABS_FRAMES	FORWARD	SPEED
ANIMATION	FRAME	STATE
AT	FROM	STATUS
AUDIO	ID	TIME
AUDIO_BOTH	IN	TIMEOUT
AUDIO_LT	INPUT	TO
AUDIO_RT	INTO	TRACK
BY	LEVEL	UP
CHANNEL	LIST	UP_AT
COLUMN	MASTER	UP _IN
CUE	OFF	UNTIL
DEVICE	ON	VIDEO
DOWN	OUTPUT	WITH
DOWN _AT	POSITION	

3.1.7 Parameters

Parameters are used in conjunction with TOKENs. Each verb has different parameters that may be acceptable, depending on its use. They may be one of the following:

- ☑ Variable
- ☑ Logical Input Channel
- ☑ Logical ObjectChannel
- ☑ Cue Number Address
- ☑ Animation Number Address
- ☑ Time, denoted - hh:mm:ss:ff
- ☑ Number
- ☑ Alias
- ☑ Label

3.1.8 Labels

Labels are used in conjunction with the GOTO Verb. A label is defined by entering the label text followed by a colon. The label must begin a valid event line. You can then jump to that event by using **Goto** followed by the label text.

The following example will loop a Laser Disk Player:

LABEL	VERB	OBJECT	Token 1	Param 1
Start:	Seek	'PRESHOW_LDP'	to 1000;	
	delay		30;	
	Play	'PRESHOW_LDP';		
	delay			00:00:30.00;
	goto	Start;		

Seeks LDP to frame 1000

Waits for 1 second

Starts Player

Delays for 30 seconds

Goes back and runs again

3.1.9 Strings

Strings are supported in two ways. A string can be sent to the monitor port by simply using a DISPLAY verb with your string enclosed in double quotes. This is typically destined to the HMT 4010 Hand Held Maintenance Terminal.

DISPLAY "Show ABC Started" row 0 column 0;

This example sends the string to row 0 column 0 of the display.

The second usage of strings is with the SEND verb. This verb is used when a literal string needs to be sent to a serial device defined on any serial port. It can be used to send special codes or characters to a device that may not be defined in the device protocol.

The verb SEND takes a normal object parameter. The maximum string length for one event line is 75 characters. The string may be formatted with hex expressions for non-ASCII codes. The following example demonstrates this string type:

Send 'SERIAL_DEVICE' "This string with a \x02 code";

3.2 Programming Reference

This section provides detailed explanation of the commands (verbs) used in the Media Pro® 4000 software. Each verb is described here first by the module and second, by alphabetic order. Most of the commands are used with the ICM, however, some commands have very specific structure for use with other modules. Refer to the Module's User Manual for the verb usage particular to that module.

3.2.1 ICM – Intelligent Control Module

The ICM provides the control, memory and communications functions for the Media Pro® 4000 system. Each module contains personality information that is stored in non-volatile Flash Memory on each unit. This memory is divided into three sections: Module Based Code, Parameters and Show Specific Data (if applicable). This feature allows for easy field system ***extensibility.***

The ***Module Based Code*** memory is that area which contains the on-board processor executable software for the specific module. This software dictates the module functionality and can be easily upgraded in the field via a PC or Laptop using a diagnostic download utility. This means that as code improvements or functional enhancements are required, they can be done by the user and do not have to be removed and returned to the factory. System updates are available via CompuServe or the Internet.

The ***Parameters Based Data*** memory is used to store user settings of options provided by the specific module. Communications ports configuration is stored in this region. All parameters are set by the Media Pro® Software utilities and downloaded from a PC or Laptop.

The ***Specific Show Data*** memory contains an actual application resource. These resources can be thought of like a video laser disk. Each resource can be played, stopped, stepped, etc. Depending on the type of resource, it could be coming from a DSM card in the form of an Audio Sound, from an ICM memory for Animated Figures, or DMX Data from a LCM to control lighting. In all cases, the control of these devices is done with the same Media Pro® Control Language (MPCL) used in the Media Pro® Cues.

3.2.2 Module Programmable Functions

- ❑ Communication Protocols
- ❑ Baud Rates
- ❑ Data Bits
- ❑ Stop Bits
- ❑ Parity

Depending on the protocol, additional settings may be present (for example, the Allen-Bradley DF1 protocol requires additional information relating to the PLC address, the starting data address to transfer, the amount of data to transfer, the related Media Pro® addresses, etc.).

3.3 Command Formats and Examples

ABS

Sets an object to an absolute value.

(optional)	*required*	*required*		*as needed - 32 bit number (object could be used as a token)*				
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Abs	@v,@l,@o						

AND

Combines objects together.

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	And	@v,@i,@o	With	@v,@i,@o,#				

DECREASE

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Decrease	@v,@i,@o	By	@v,@i,@o,#				

DELAY

This command causes the cue to be inactive for the specified amount of time.

(optional)	*required*	*required*		*as needed - 32 bit number (object could be used as a token)*				
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Delay			hh:mm:ss.ff				
	Delay			@v,@i,@o,#				

DISPLAY

Used to send a message to the CRT of the Hand-held Maintenance Module connected to an ICM via a serial port.

(optional)	*required*	*required*		*as needed - 32 bit number (object could be used as a token)*				
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Display	“String”	Row	#	Column	#		
	Display	“String”						

The following is a generic description of the DISPLAY command:

Display Statements Used in Media Pro Cues -

1 - Display Statements
   A) Display "string";
      1) Display the text in-between the "quotes".
      2) Display ""; clears the current line.
   B) Display "string" Row 0 Column 0;
      1) Displays the text in-between the "quotes" on row 0 starting at column 0.
      2) The range for row and column is limited by the VT-52 emulator the display command is sent to.
         (a) Typically the display on the ICM cue edit dialog in the MP4000 software, or the Hand Held Maintenance Terminal. (Rows 0 through 3, Columns 0 through 19)

      Example - Display "" Row 0 Column 0;
      Clears line 0.

2 - Variables, Input Channels, and Output Channels can be put in the display string.
   A) Variables use %d###.bb, %d###, %D###, %h###, %H###, %t###, %T###.
      1) Where ### is the Variable Number (0-511)
      2) Where d & D stand for Decimal Format
         (a) d & D are equivalent
         (b) d & D will display the value less than 65535 in decimal format.
      3) Where h & H stand for Hexadecimal Format
         (a) h will display the value in 16 bit hexadecimal format (0000-FFFF).
         (b) H will display the value in 32 bit hexadecimal format (00000000-FFFFFFFF).
      4) Where t & T stand for Timecode Format
         (a) t & T are equivalent
         (b) t & T will display the value in Time Code Format hh:mm:ss.ff, for values less than 24:00:00.00.
      5) Where bb is a specific bit (0-31) of the Variable

      Example - Display "%H123";
      Would display Variable Number 123's value in Hexadecimal.

      Example - Display "%D123.31";
      Would display Variable Number 123, bit 31's value as 0 or 1.

DISPLAY (continued)

B) Input Channels use %id####.bb, %id####, %iD####, %ih####, %iH####, or %Id####.bb, %Id####, %ID####, %Ih####, %IH####.

1) Where #### is the Input Channel (0-1023)
2) Where d & D stand for Decimal Format
   (a) d & D are equivalent
   (b) d & D will display the value less than 65535 in decimal format.
3) Where h & H stand for Hexadecimal Format
   (a) h will display the value in 16 bit hexadecimal format (0000-FFFF).
   (b) H will display the value in 32 bit hexadecimal format (00000000-FFFFFFFF).
4) Where bb is a specific bit (0-7) of the Input Channel

Example - Display "%iH123";
Would display Input Channel 123's value in Hexadecimal.

Example - Display "%iD123.7";
Would display Input Channel 123, bit 7's value as 0 or 1.

C) Output Channels use %od####.bb, %od####, %oD####, %oh####, %oH####, or %Od####.bb, %Od####, %OD####, %Oh####, %OH####.

1) Where #### is the Output Channel (0-1023)
2) Where d & D stand for Decimal Format
   (a) d & D are equivalent
   (b) d & D will display the value less than 65535 in decimal format.
3) Where h & H stand for Hexadecimal Format
   (a) h will display the value in 16 bit hexadecimal format (0000-FFFF).
   (b) H will display the value in 32 bit hexadecimal format (00000000-FFFFFFFF).
4) Where bb is a specific bit (0-7) of the Output Channel

Example Display "%oH123";
Would display Output Channel 123's value in Hexadecimal.

Example Display "%oD123.7";
Would display Output Channel 123, bit 7's value as 0 or 1.

Examples -

(a) Hours running in time code display.
   (1) Display "" Row 0 Column 0;
   (2) Display "" Row 1 Column 0;
   (3) Display "Hrs Running" Row 0 Column 0;
   (4) Display "%T123" Row 1 Column 0;
(b) The statements above will clear rows 0, and 1, then Display Hrs Running on Row 0, and the value in variable 123 in Timecode format on row 1.
   (1) Where the Value of Variable 123 is 2,591,999.
   (2) looks like: Hrs Running
   23:59:59.29

DISPLAY (continued)

Example - Output channels

(c) Display "ch12:%oD12, ch13:%oD13";

(1) Where channels 12 and 13 are 255 decimal.

(2) looks like: ch12:255, ch13:255

Example - Input Channel Bits

(d) Display "ch800:%id800.2%id800.1%id800.0";

(1) Where channel 800 has bits 0, and 2 on, and bit 1 off.

(2) looks like: ch800:101

DIVIDE

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Divide	@v,@i,@o	By	@v,@i,@o,#				

GOTO

This verb will cause the cue to jump to the sub-routine defined by the object defined as LABEL. OBJECTS are defined in the cue .

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Goto	Label						

IF

This verb will compare the values defined and branch to a subroutine when they are true.

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	If	@v,@i,@o	!=	@v,@i,@o,#	Goto	Label		
	If	@v,@i,@o	<	@v,@i,@o,#	Goto	Label		
	If	@v,@i,@o	<=	@v,@i,@o,#	Goto	Label		
	If	@v,@i,@o	=	@v,@i,@o,#	Goto	Label		
	If	@v,@i,@o	>	@v,@i,@o,#	Goto	Label		
	If	@v,@i,@o	>=	@v,@i,@o,#	Goto	Label		

INCREASE

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Increase	@v,@i,@o	By	@v,@i,@o,#				

INVERT

Inverts the specified data.

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Invert	@v,@i,@o						

LOAD

Loads a resource from a device. There are examples of specific Load Device and Load List commands for use with particular serial protocols. They can be found in the Programming Section of the ICM, SEM, and IMC manuals.

The following is a list of generic Load statements:

(optional)	*required*	*required*		*as needed - 32 bit number (object could be used as a token)*				
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Load	@r	Device	@v,@i,@o,#	Cue	@v,#		
	Load	@r	Device	@v,@i,@o,#	Disk	@v,#		
	Load	@r	Device	@v,@i,@o,#	List	@v,#		
	Load	@r	Device	@v,@i,@o,#	Path	@v,#		
	Load	@r	List	@v,@i,@o,#	Patch	@v,#		
	Load	@r						
	Load	@v	With	@r	Status	n	Byte	b-e
	Load	@v	From	@r	Status	n	Byte	b-e

Cue Control with Status –

Use of status for cue control (play/stop/reset) is indirect. As outlined below, the LOAD statement permits attaching a status to a variable, input byte or output byte, and these are then used for cue control.

Load Status Statement –

The Load Status statement is a means for Cue statements to get status information from modules in the system. The LOAD FROM statement requires ICM Firmware 3220119U.COD (04-06-00) or later, ***and*** MP4000 Software 1.9.1.9 (04-15-00) or later. Refer to "Downloading New Firmware to a Module" in section 6, if firmware needs to be updated.

The following is a list of generic examples of the Load Status statements:

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Load	@Vvvv	With	@Rr,s[p]	Status	n	Byte	b-e
	Load	@Iiii	With	@Rr,s[p]	Status	n	Byte	b-e
	Load	@Oooo	With	@Rr,s[p]	Status	n	Byte	b-e
	Load	@Vvvv	With	@Aaaa	Status	n	Byte	b-e
	Load	@Vvvv	With	@Qqqq	Status	n	Byte	b-e
	Load	'Alias_1'	With	'Alias_2'	Status	n	Byte	b-e
	Load	@Vvvv	From	@Rr,s[p]	Status	n	Byte	b-e
	Load	@Iiii	From	@Rr,s[p]	Status	n	Byte	b-e
	Load	@Oooo	From	@Rr,s[p]	Status	n	Byte	b-e
	Load	@Vvvv	From	@Aaaa	Status	n	Byte	b-e
	Load	@Vvvv	From	@Qqqq	Status	n	Byte	b-e
	Load	'Alias_1'	From	'Alias_2'	Status	n	Byte	b-e

- Where [@Vvvv,@Iiii,@Oooo]
  - selects the destination for the status (in the 'syntax' pull-down menu in MP4000, this is [anyv]).

- Where @Rr,s,[,p]
  - selects the source of the status
    - In the source, if the port [,p] is not given, the status will refer to the module rather than one of its ports.

- The value n is the 'index' to the status-of-interest in the set associated with the specified port @Rr,s,p or module @Rr,s. The various indices available for each module are listed in the Module's User Manual.
  - The status item size may be 8, 16, or 32 bits, as detailed in the module manuals, along with the description of the corresponding status.
  - Using 0 for n always retrieves a set of status bits for the source.
    - Values 1 and higher are for data specific to the port or module.
  - Some status items may be arrays of identically sized values –
    - e.g: LCM4020 item 6, DMX data
  - If n is a number higher than the highest status available, 0's will be returned.
    - The status-of-interest available for each module is defined in Section 5 of its own User Manual.

❖ The statement format permits specifying which bytes of a status item are of interest.

- Where the range b-e specifies the first and last bytes of the range to move.
  - Ordinarily, a BYTE range is used that corresponds exactly with the size of the status:

    8-bit status: BYTE 0-0
    16-bit status: BYTE 0-1
    32-bit status: BYTE 0-1
    Arrays of 8-bit: BYTE b-e
  - Where 0 is the least significant Byte.
    - If the range selected is larger than the value returned, the upper remaining bytes will be set to 0's.
    - If the range selected is larger than the destination, the next object(s) will be filled with 0's also.
    - If the status byte is not fully implemented, it will return 0 or other invalid values.
  - The BYTE token and b-e specifier are a way to control and/or limit what status is moved.
    - Examples: You may move the second byte of a long (4-byte) status into an input block byte, by using:
      - LOAD @I200 WITH @R0,17,0 STATUS 0 BYTE 1-1 ;
    - You may load 16 DMX channels from the LCM card into 4 consecutive variables by using
      - LOAD @V100 WITH @R0,5 STATUS 6 BYTE 20-35

    Example: Load @i123 with @Rr,s,p Status 0 Byte 0-3;

    Fills @i123 with Byte 0
    Fills @i124 with Byte 1
    Fills @i125 with Byte 2
    Fills @i126 with Byte 3

- Where "WITH"
  - Establishes a one-time request

- Where "FROM"
  - Establishes a persistent delivery to the target

Single use status is most useful (and least impacting on system bandwidth) for status that is only needed occasionally. Persistent status is particularly useful where some status must be used to trigger cues, as described in the next section -- the persistent update then permits the cue triggering without any 'maintenance' cue constantly running.

Note: Persistent status is set up by cue statements - it is **not** part of a rack's configuration. If a rack is reset manually, by a download of code or configuration, or by a power cycle, all the persistent status setup will be lost, and the master rack's cues will need to be re-run to provide the required direction to the slave racks.

PERSISTENT STATUS TIMING CONSIDERATIONS

The LOAD/FROM statement establishes a 'connection' from the specified status to the selected object (variable or byte). Once a LOAD statement of this sort has been issued, the selected object is updated routinely and at change by the ICM, over the network if the status item is remote.

For items outside the master rack, there is some delay in initiating the update sequence, so a DELAY will be needed after LOAD statements. The delay ranges from about 2 frames for systems with two or three slave racks, to about 10 frames for systems with a full complement of 31 racks.

[This recommended/necessary delay applies also to single-report LOAD/WITH statements sent to slaves, since the messaging operations are the same, except that the trigger is shut down after a single use.]

After the routine update is established, the item will be updated within a frame of its change in small systems (four or less slaves) or within three to four frames in large systems (twenty-five or more slaves).

This update rate may be affected if many status items from remote racks are LOADed. There is no delay for items in the master rack; in fact, the destination is updated for the first time immediately, as part of the execution of the LOAD statement.

Each rack, master and slaves, can have up to 500 active status items, counting both persistent and any instantaneous load of single-report requests for status.

Slave racks will transmit the value of status when the request is initiated, and then again whenever the status changes. The comparison with a previous value, inhibiting messaging if there is not a change of value, can only be done for status values of 4 bytes or less. If a selection from an array is made that is more than 4 bytes, the comparison will be overridden, and the status values will be transmitted to the master in *EVERY* frame.

An UNLOAD, or a LOAD/WITH, or another LOAD/FROM command, to a destination of a persistent status, will cancel or modify the older 'connection'. There are comments about the UNLOAD command below.

If a destination is one of several involved in a persistent update (perhaps initiated with a wide byte-range on an array of data), then a LOAD to that destination will disconnect the *entire* original range in which it was included. Similarly, a new LOAD instruction to the same (or any contained) destination will cancel any previous connections to that destination.

UNLOAD STATUS

The following statement releases the variable, input byte or output byte for normal use:

UNLOAD [@Vv,@Ii,@Oo] ;

For more specific information regarding the Unload statement, refer to the Unload page further in this section.

DATA MOVED, MORE ABOUT "BYTE b-e"

The amount of data (the number of bytes) moved to the selected object is ordinarily the size of the specified data, even if a byte destination is selected. For example, an analog 'following error' would be typically two bytes. If LOADed to a variable, those two bytes are moved to the least significant two bytes of the variable, and the other two bytes of the variable are set to zero. If that following error is LOADed to an output byte, both bytes are moved: the least significant byte to the specified output byte, and the other byte to the NEXT output byte in order.

The required BYTE b-e argument is used to specify a number of bytes to use from the specified status item. The b-e 'range' parameter gives the first (beginning) and last (ending) byte of the status item to move to the destination. These first/last values must be within the specified status item. Bytes are numbered 0-based, so an 8-bit item has byte 0, a 16-bit status item has bytes 0 and 1, and a 32 bit status item has bytes 0, 1, 2, and 3. For example, to pick out the second byte from a DSM's port status that contains the 'playing' bit, use

LOAD @I710 WITH @R0,3,2 STATUS 0 BYTE 1-1;

The bit is then tested with IF @I710.4 = 1 GOTO Playing; . Note that these examples are made up; use the module's manual to find the exact status item, byte, and bit to use.

A status item that is an array will have many bytes. To get some subset of the array requires that the BYTE parameter have the appropriate byte range specified. For example, the LCM status contains an array of the output DMX data. To access, say, lighting channels 4 through 12, this statement would be used:

LOAD @V32 WITH @R0,2 STATUS 6 BYTES 4-12;

This statement would map:

- lighting channels **4,5,6,7** to variable 32
- lighting channels **8, 9,10,11** to variable 33
- lighting channel **12** to variable 34.

STATUS vs. FRONT PANEL SWITCH

If the master rack's front panel switch (on the ICM) is placed in the TEST position, *ALL* status reporting from the master *and* any slaves is stopped. The setups are not lost, so the reporting will resume when the switch is returned to the RUN or PROG positions.

If a slave rack's front panel switch (on the ICM) is placed in the TEST position, ALL status reporting from *that* slave is stopped. The setups are not lost, so the reporting will resume when the switch is returned to the RUN or PROG positions.

MULTIPLE DESTINATIONS

Note that the same status may be LOADed to more than one destination. The maximum number of LOAD statements that can be active at any one time is 500 in every rack. The 'connections' can be UNLOADed and reLOADed dynamically as necessary. See UNLOAD, below.

BAD SPEC HANDLING (not fully implemented)

If the index specifier (n) refers to a status item not defined by the module's header, the selected object will NOT be modified, and the 'connection' will not be established. A count of such attempts is kept and may be accessed by using the HMT or by using LOAD statements. A count is also kept of the number of active LOAD statement connections. See the description of ICM Module Status items, located in the ICM User Manual, Section 5.8, for those counts.

APPLICATION NOTE

It may be useful sometimes to retrieve only a portion of several bytes available. For example, in the 'download version' there are four bytes; the download number is in the least significant byte, and the download letter is in the most significant byte. For example, the download version 3220119U.COD is returned into a variable as (hex) 55000077. It may be easier to deal with in a cue if loaded into a byte variable with a 0-0 range, giving the (hex) 77 value, 119 decimal; or a 3-3 range, giving the(hex) 55 value (an ASCII 'U').

Suppose you wish, at different times, to retrieve a different set of elements of a status array, for example, the LCM's DMX data. You can use a variable in the BYTE specification and reduce the number of statements in your cues. The following cue gets all the 512 DMX bytes from the LCM, one at a time, and sends them to the HMT:

```
SET @V99 TO 0 ;
SET @V100 TO 1-1 ;
LOOP: LOAD @V101 WITH @R0,5 STATUS 6 BYTE @V100 ;
DISPLAY "%d101" ;
INCREASE @V100 BY 1-1 ;  V100 becomes 2-2, 3-3, etc.
INCREASE @V99 BY 1 ;
IF @V99 < 512 GOTO TOP ;
```

MULTIPLY

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Multiply	@v,@i,@o	By	@v,@i,@o,#				

NEGATE

(optional)	*required*	*required*	*as needed - 32 bit number (object could be used as a token)*					
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Negate	@v,@i,@o						

OR

(optional)	*required*	*required*		*as needed - 32 bit number (object could be used as a token)*				
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Or	@v,@i,@o	With	@v,@i,@o,#				

PLAY

Tells the resource to begin at the specified place or cue, depending on the tokens and parameters used. All configurations can be used with the ICM and SEM modules:

(optional) required required as needed - 32 bit number (object could be used as a token))

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Play	@q						
	Play	@r						
	Play	@r	Device	@v,@i,@o,#	Cue	@v,#	List	@v,#
	Play	@r	Device	@v,@i,@o,#				
	Play	@r	Device	@v,@i,@o,#	Disk	@v,#		
	Play	@r	Device	@v,@i,@o,#	Track	@v,#		
	Play	@r	Device	@v,@i,@o,#	Disk	@v,#	Track	@v,#
	Play	@r	Disk	@v,@i,@o,#	Track	@v,#		
	Play	@r	From	@v,@i,@o,#	To	@v,#		

The following is a generic description of the PLAY command:

PLAY @Rr,s,p ;

- Where r,s,p is the rack, slot, and port address of the module.

This statement resumes operation of currently active cue if STOPed; or, if the active cue is 'complete', makes the pending cue active and plays it. Note that PLAY has no effect on an active cue that is already 'playing'. It will not immediately execute the link in the active cue – see STEP, command.

Additional description of the PLAY command structure for use with the SEM:

PLAY 'db' DEVICE ua-ty TRACK tk-sk SEGMENT sn ;

- where 'db' is the alias or @Rr,s,p designation of the serial port
  - ua-ty is the hyphenated unit address (1-based) and type
  - tk-sk is the reproducer number (1-based) and socket ('mode')
  - sn is the sound number or video number on the reproducer.
- generates F0 7F id 02 7F 01 T1 T0 00 SK 00 S2 S1 S0 F7 for type 1
- or F0 7F id 02 7F 03 T1 T0 00 SK 00 S2 S1 S0 F7 for type 2
  - id is either 7F (if ua is 127) or ua + 8 (since ua is 1-based).
  - T1/T0 is a two ascii-digit sequence giving tk. (For instance, if tk is 20, this short string is "20".)
  - SK is an ascii-digit giving the value of sk, i.e., 1 -> "1".
  - S2/S1/S0 is a three ascii-digit sequence giving the value of sn;
  - for the Digital Binloop the range is "001" to "511". These are direct translations of sn, no offsets.

RAMP

This verb is used to tell a device to change to a specific level.

(optional)	*required*	*required*		*as needed - 32 bit number (object could be used as a token)*				
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Ramp	@r	Down	@v,@i,@o,#				
	Ramp	@r	Down_At	@v,@i,@o,#				
	Ramp	@r	Down_In	@v,@i,@o,#				
	Ramp	@r	To	@v,@i,@o,#				
	Ramp	@r	To	@v,@i,@o,#				
	Ramp	@r	Up	@v,@i,@o,#				
	Ramp	@r	Up_At	@v,@i,@o,#				
	Ramp	@r	Up_In	@v,@i,@o,#				

The following is a generic description of the RAMP command:

Ramp @"r,s,p" channel "ch#" To "%" In "time"

- Where r,s,p is the rack, slot, and port address, ch represents the channel number or range, and time is a value in tenths of seconds.

More specific explanation of the RAMP command, used for the LCM follows:

RAMP @Rr,s,p CHANNEL [cc, bc-ec] TO pp IN tt ;

- Where **cc** is a single channel specification
- **bc-ec** is a range specification (begin channel to end channel, inclusive)
- **pp** is a percentage 0-100 of full on
- **tt** is a time expressed in 10th of seconds (even though the internal ramp rate is 30ths of seconds) from 0 to 99.9 seconds.

RANDOM

Tells the device to select the resources randomly.

(optional)	*required*	*required*	*as needed - 32 bit number (object could be used as a token)*					
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Random	@v,@i,@o	From	@v,@i,@o,#	To	@v,#		

Typical Examples of the Random Verb usage:

1 - Example Syntax -
- A) Random @v1 From 0 to 65534;
  1) Puts a random # between 0 and 65534 into Variable 1
- B) Random @v2 From 1 to 65535;
  1) Puts a random # between 1 and 65535 into Variable 2
- C) Random @v3 From @v1 to @v2;
  1) Puts a random # between @v1 and @v2 into Variable 3
- D) Random @i800 From 0 to 255;
  1) Puts a random # between 0 and 255 into input channel 800
- E) Random @o600 From 0 to 255;
  1) Puts a random # between 0 and 255 into output channel 600
- F) Random @v63 From 65534 to 2;
  1) Generates one of the #s (65534, 65535, 0, 1, 2)

2 - Example Cue

```
LOOP: Random @v10 from 1 to 3; Puts a Random # between 1 and 3 into Var 10
If @v10 = 1 GoTo CRASH; If Variable 10 equals 1 Play the Crash Sequence
If @v10 = 2 GoTo BANG; If Variable 10 equals 2 Play the Bang Sequence
If @v10 = 3 GoTo BOOM; If Variable 10 equals 3 Play the Boom Sequence
GoTo LOOP;
CRASH: Play 'DSM01' Track 1 Segment 1;  Plays Crash Sound Effect
Delay 00:00:05.00; Delay 5 seconds for Crash Sequence to complete
GoTo LOOP;
;
BANG: Play 'DSM01' Track 1 Segment 2;  Plays Bang Sound Effect
Delay 00:00:01.00; Delay 1 seconds for Bang Sequence to complete
GoTo LOOP;
;
BOOM: Play 'DSM01' Track 1 Segment 3;  Plays Boom Sound Effect
Delay 00:00:02.00; Delay 2 seconds for Boom Sequence to complete
GoTo LOOP;
```

3 - Notes -
1) Random Generates a Integer Random Number.
2) The Difference Between From and To must be less than or equal to 65534.
3) The Random # can be placed into a Variable, Input Channel, or Output Channel.
4) The Random # Verb can be passed Constants, or Variables in the FROM & TO tokens.
5) The Random numbers will not repeat on a hard, or soft reset.
6) The Random numbers will repeat at power on, If a real time clock is not installed.

RECORD

All formats of this verb can be used with the ICM and SEM:

(optional)	*required*	*required*	*as needed - 32 bit number (object could be used as a token)*					
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Record	@r	From	@v,@i,@o,#	To	@v,#		
	Record	@r	From	@v,@i,@o,#				
	Record	@r	To	@v,@i,@o,#	In	@v,#		
	Record	@r	To	@v,@i,@o,#				
	Record	@r	Track	@v,@i,@o,#	Segment	@v,#		
	Record	@r	Track	@v,@i,@o,#	Segment	@v,#	Time	@v,#
	Record	@r	Track	@v,@i,@o,#				
	Record	@r						

RESET

(optional)	*required*	*required*	*as needed - 32 bit number (object could be used as a token)*					
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Reset	@r	Device	@v,@i,@o,#				
	Reset	@q,@r						

The following is a generic description of the RESET command:

RESET @Rr,s,p ;

- Where @Rr,s,p is the rack/slot/port location of the port to be affected.

This statement causes all the channels in a port to be set to 0, there is no active cue and no pending cue. If any channels for the port are 'animated', the animation data will have effect at the beginning of the next frame.

RESUME

Tells the device or cue to continue.

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Resume		Device	@v,@i,@o,#	Cue	@v,#		
	Resume		Device	@v,@i,@o,#	Cue	@v,#	List	@v,#
	Resume		Device	@v,@i,@o,#				

SEEK

Tells the device to go to a specified location.

(optional) required required as needed - 32 bit number (object could be used as a token))

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Seek	@r	To	@v,@l,@o,#				
	Seek	@r	Device	@v,@i,@o,#	Track	@v,#		
	Seek	@r	Track	@v,@i,@o,#	Segment	@v,#		
	Seek	@r	Track	@v,@i,@o,#	Segment	@v,#	Time	@v,#
	Seek	@r	Track	@v,@i,@o,#	To	@v,#		
	Seek	@r	Track	@v,@i,@o,#				

The following is a generic description of the SEEK command:

When the MP4000 compiler operates on a statement like:

```
SEEK 'DR8' TO 00:01:15.00 ;
```

The hours/minutes/seconds/frames format is converted to an absolute frame number for download into the ICM. This conversion is based on the assigned rate for the cue, selected in the 'Rate' drop down window, one of 24, 25, 29.97, or 30 fps.

For example, the above specification would be:

- at 24 fps: 1800 frames,
- at 25 fps: 1875 frames,
- at 30 fps: 2250 frames.

SEND

Sends a string of information to the specified object.

(optional)	*required*	*required*		*as needed - 32 bit number (object could be used as a token)*				
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Send	@r		"String"				

1 - Send Statements
- A) Send 'Alias' "string";
  1) Sends the text in-between the "quotes" to the serial port that is assigned the alias 'Alias'.
  2) The @rRack,Slot,Port object can also be used.
     (a) Send @r0,17,2 "string";
- B) Send 'Alias' "Carriage Return\x0D";
  1) ASCII codes can be put in the string using a \ or \x, followed by the ASCII character in Hexadecimal.
  2) Carriage Return is \x0D, Line Feed is \x0A
  3) ASCII Chart

HEX	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
00	NUL	01	SOH	02	STX	03	ETX
04	EOT	05	ENQ	06	ACK	07	BEL
08	BS	09	HT	0A	LF	0B	VT
0C	FF	0D	CR	0E	SO	0F	SI
10	DLE	11	DC1	12	DC2	13	DC3
14	DC4	15	NAK	16	SYN	17	ETB
18	CAN	19	EM	1A	SUB	1B	ESC
1C	FS	1D	GS	1E	RS	1F	US
20	space	21	!	22	"	23	#
24	$	25	%	26	&	27	'
28	(	29	)	2A	*	2B	+
2C	,	2D	-	2E	.	2F	/
30	0	31	1	32	2	33	3
34	4	35	5	36	6	37	7
38	8	39	9	3A	:	3B	;
3C	<	3D	=	3E	>	3F	?
40	@	41	A	42	B	43	C
44	D	45	E	46	F	47	G
48	H	49	I	4A	J	4B	K
4C	L	4D	M	4E	N	4F	O
50	P	51	Q	52	R	53	S
54	T	55	U	56	V	57	W
58	X	59	Y	5A	Z	5B	[
5C	\	5D	]	5E	^	5F	_
60	`	61	a	62	b	63	c
64	d	65	e	66	f	67	g
68	h	69	i	6A	j	6B	k
6C	l	6D	m	6E	n	6F	o
70	o	71	q	72	r	73	s
74	t	75	u	76	v	77	w
78	x	79	y	7A	z	7B	{
7C	|	7D	}	7E	~	7F	DEL

C) Variables can be put in the send string.
   1) Variables use %d###.bb, %d###, %D###, %t###, %T###.
      (a) Where ### is the Variable Number (0-511)
      (b) Where d & D stand for Decimal Format
         (1) d & D are equivalent
         (2) d & D will send the valueless than 65535 in decimal format.
      (c) Where t & T stand for Timecode Format
         (1) t & T are equivalent
         (2) t & T will send the value in Time Code Format hh:mm:ss.ff, for values less than 24:00:00.00.
      (d) Where bb is a specific bit (0-31) of the Variable

   Example - Send 'Alias' "%H123";
   Would display Variable Number 123's value in Hexadecimal.

   Example - Send 'Alias' "%D123.31";
   Would display Variable Number 123, bit 31's value as 0 or 1.

SET

Tells the device to set to a specific or relative level.

(optional) required required as needed - 32 bit number (object could be used as a token))

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Set	@r	Abs_Frames	@v,@i,@o,#				
	Set	@r	Position	@v,@i,@o,#				
	Set	@r	Rate	@v,@i,@o,#	SmpteTime	@v,#		
	Set	@r	Video	@v,@i,@o,#				
	Set	@r	Audio_Both	@v,@i,@o,#				
	Set	@r	Audio_Left	@v,@i,@o,#				
	Set	@r	Audio_Right	@v,@i,@o,#				
	Set	@r	DbLevel	@v,@i,@o,#				
	Set	@r	Dev_Disk	@v,@i,@o,#				
	Set	@r	Dev_Frame	@v,@i,@o,#				
	Set	@r	Dev_Segment	@v,@i,@o,#				
	Set	@r	Dev_Stat	@v,@i,@o,#				
	Set	@r	Dev_Track	@v,@i,@o,#				
	Set	@r	Device	@v,@i,@o,#	Audio	@v,#		
	Set	@r	Device	@v,@i,@o,#	DbLevel	@v,#	In	@v,#
	Set	@r	Device	@v,@i,@o,#	DbLevel	@v,#		
	Set	@r	Device	@v,@i,@o,#	Fade_In	@v,#		
	Set	@r	Device	@v,@i,@o,#	Fade_Out	@v,#		
	Set	@r	Device	@v,@i,@o,#	Fader	@v,#		
	Set	@r	Device	@v,@i,@o,#	Item	@v,#		
	Set	@r	Device	@v,@i,@o,#	Level	@v,#	In	@v,#
	Set	@r	Device	@v,@i,@o,#	Level	@v,#		
	Set	@r	Device	@v,@i,@o,#	Master	@v,#		
	Set	@r	Device	@v,@i,@o,#	Rate	@v,#		
	Set	@r	Display	@v,@i,@o,#				
	Set	@r	Fade_In	@v,@i,@o,#				
	Set	@r	Fade_Out	@v,@i,@o,#				
	Set	@r	Fader	@v,@i,@o,#				
	Set	@r	Id	@v,@i,@o,#				
	Set	@r	Input	@v,@i,@o,#	DbLevel	@v,#		
	Set	@r	Input	@v,@i,@o,#	Output	@v,#		
	Set	@r	Input	@v,@i,@o,#				
	Set	@r	Item	@v,@i,@o,#	Item	@v,#		
	Set	@r	Level	@v,@i,@o,#				
	Set	@r	Master	@v,@i,@o,#				
	Set	@r	Patch	@v,@i,@o,#				
	Set	@r	Patch	@v,@i,@o,#				
	Set	@r	Rel_Frames	@v,@i,@o,#				
	Set	@r	Set	@v,@i,@o,#				
	Set	@r	Speed	@v,@i,@o,#				
	Set	@r	State	@v,@i,@o,#				
	Set	@r	Timeout	@v,@i,@o,#				

These SET verbs are used within a CUE:

(optional) required required as needed - 32 bit number (object could be used as a token))

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Set	@q,@r	Frame	@v,@i,@o,#				
	Set	@q,@r	Item	@v,@i,@o,#	Segment	@v,#	Level	@v,#
	Set	@q	Rate	@v,@i,@o,#	Timeout	@v,#	GoTo	Label
	Set	@q,@r	Rate	@v,@i,@o,#				
	Set	@r,@i,@o,@v	To	@v,@i,@o,#				

STEP

(optional)	*required*	*required*	*as needed - 32 bit number (object could be used as a token)*					
LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Step	@r	Device	@v,@i,@o,#	Cue	@v,#		
	Step	@r	Device	@v,@i,@o,#	Cue	@v,#	Reverse	@v,#
	Step	@r	Device	@v,@i,@o,#	List	@v,#		
	Step	@r	Device	@v,@i,@o,#	List	@v,#	Reverse	@v,#
	Step	@r	Device	@v,@i,@o,#	Reverse	@v,#		
	Step	@r	Device	@v,@i,@o,#				
	Step	@r	Forward	@v,@i,@o,#				
	Step	@r	Reverse	@v,@i,@o,#				

The following is a generic description of the STEP command:

STEP @Rr,s,p

- Where r,s,p is the rack, slot, and port address of the module

This command will play the pending cue; this works even if the port is currently playing a cue. The pending cue is either the linked cue from the current cue, or the cue in next numerical order. Note that the last cue (i.e., with the highest number) for the port has no 'next' cue unless it has a link.

Additional command information for the STEP command usage:

STEP @Rr,s,p REVERSE

This command will play the previous cue; the selection is NOT in order of play, reversed, but the cue with the numerical sequence less than the current cue is determined and played.

STOP

Commands a device to Stop.

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Stop	@r,@q						
	Stop	@r	At	@v,@i,@o,#				
	Stop	@r	Device	@v,@i,@o,#	Cue	@v,#		
	Stop	@r	Device	@v,@i,@o,#	Cue	@v,#	List	@v,#
	Stop	@r	Device	@v,@i,@o,#				

UNLOAD

This command directs the device to unload a resource:

(optional) required required as needed - 32 bit number (object could be used as a token))

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Unload	@r	Device	@v,@i,@o,#				
	Unload	@r	List	@v,@i,@o,#	Patch			
	Unload	@r						

Unloading the persistent update of status is:

```
Unload @Vvvv;
Unload @Iiii;
Unload @Oooo;
Unload 'Alias';
```

The UNLOAD statement merely signals the system that the continual update is no longer needed. This is essentially a cleanup issue to help keep the net usage down. If a variable or byte is being reconnected from one status item to another status item, it is not necessary to use UNLOAD; the newly issued LOAD statement will effect the disconnection before initiating the new update sequence.

WAIT

Instructs the cue to wait a specified length of time or number of frames.

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Wait			hh:mm:ss.ff				
	Wait			@v,@i,@o,#				

XOR

(optional) required required as needed - 32 bit number (object could be used as a token)

LABEL:	VERB	OBJECT	Token 1	Param 1	Token 2	Param 2	Token 3	Param 3
	Xor	@v,@i,@o	With	@v,@i,@o,#				

3.4 Example ICM Program

If the ICM Port 0 were assigned the Protocol; Pioneer Laser Disk Player and the Alias 'LDP01,' a few commands would be:

- ❏ Load 'LDP01' - (Spins up the Laser Disk)
- ❏ Seek 'LDP01' to 1000 – (Seeks a CAV Laser Disk to frame 1000)
- ❏ Play 'LDP01' – (Plays Laser Disk from Current Frame)
- ❏ Set 'LDP01' Audio 0 – (Mutes the Players Audio)
- ❏ Set 'LDP01' Audio 1 – (Enables the Players Audio)
- ❏ Set LDP01' Video 0 – (Blanks the Players Video)
- ❏ Set LDP01' Video 1 – (Enables the Players Video)
- ❏ Unload 'LDP01' – (Parks the Laser Disk)

(Some Protocols also allow strings to be sent to the device)

- ❏ Send 'LDP01' "20SPMF\x0d" – (Set the speed to 20, multi speed forward).
- ❏ Send 'LDP01' "ch3se\x0d" – (Seeks a CLV Laser Disk to chapter 3).
- ❏ Send 'LDP01' "rjco\x0d" – (Reject the Laser Disk, Close the Laser Disk Drawer, Hex value for a Carriage Return Character).

A reason to use the above SEND command instead of the UNLOAD command, is that the Pioneer Laser Disk Players will, when given the REJECT command, park the disk if the disk is playing, or open the Drawer if the disk is not playing. Following the REJECT command with a CLOSE command makes the player close its drawer if the disk was parked when given the REJECT command.

4 Section 4 – Media Pro® Tutorial

This section presents an example of the Media Pro® system usage in a typical standalone pre-show and theater. It is intended for educational purposes and may not reflect a completely functional application.

4.1 Theatre Show Example

This example system will take the reader step-by-step through the process of configuring and programming the Media Pro® 4000 System. All applications use the same steps; however, the steps used will vary depending on the show's complexity.

The steps you will be taking:

1. Define the Media Pro® modules needed for the application.
2. Configure each module's personality.
3. Download the "resource" modules with the data.
4. Animate the real-time data required for each animation scene.
5. Write Cues for the application.
6. Test the application.

STEP 1 – DEFINING THE MODULES

4.1.1 Theater Application Scenario

This application describes a typical Theater Venue with a Pre-Show and Main Theater Presentation. The "Resources" required for the application are:

1 - Pioneer Laser Disk Player for the Pre-Show Video Presentation
2 - 70 MM Film Projectors w/ DTS Sound System for the Main Theater Presentation
2 - Sets of Automatic Doors for Main Theater Entry and Exit
2 - Operator Control Consoles, 1 for the Pre-show and 1 for the Main Theater
2 - 6 Channel DMX Dimmer Packs for House Lights and Host Key Lights
2 - Locked Sound Channels for Theater Walk-in and Walk-out Music

The guests walk into the Pre-Show area from an external queue line while area walk-in music is playing in the Pre-Show area. At the far end of the space, is a Host Podium containing an Operators Control Console (OCC).

Above the Host is a Video Screen for the Pre-Show Presentation. To the guests' right are numerous automatic doors for entrance to the Main Theater. At 8 minutes before the end of the Main Theater Presentation, the Pre-Show warning light on the OCC flashes to tell him it is safe to start the Pre-Show Presentation if the House is loaded. Once the Show Start Button is pressed, the Walk-in Music fades out, the Pre-Show House Lights fade down, and the Host Key Light comes up on the Host while a brief introduction is given.

Upon the completion of the Host Introduction, the Key Light fades to black and the Video Presentation begins. At the end of the presentation, the Host Key Light Fades up, the Walk-in music fades up, and the Automatic Entry doors to the Main Theater open.

The guests proceed to the Main Theater hearing the same Walk-in music as played in the Pre-Show. When the guests are clear the Host closes the Main Theater Entry doors from the Pre-Show OCC readying the Pre-Show for the next group of guests. As the guests are filling the Main Theater the House Lights are up. The Host is at a second OCC located just inside the entry doors of the theater. Once the guests are seated and the show system is ready to start, an indicator on the OCC will begin to flash.

The Host pushes the show start button causing the Walk-in Music to fade out and the Host Key light to ramp up. The Host introduces the presentation and, upon completion, the main presentation begins. The Host Key Light and House Lights fade out and the Film Projector starts. At the end of the Film Presentation the Host Key Light, House Lights and Walk-out music ramp up allowing the Host to say goodbye just as the Theater Exit doors open.

When the theater area is empty the Exit Doors close, readying the theater for the next group of guests, and the cycle repeats.

Based on the above show description the following modules are required:

1 - VMR 4000 17 Slot Vertical Module Rack
1 - ICM 4010 Intelligent Controller Module
1 - SEM 4020 Serial Expansion Module
1 - ABM 4010 Allen-Bradley Module
2 - IOM 4020 Digital Input/Output Modules
1 - DSM 4010 Digital Sound Module
1 - LCM 4020 Lighting Control Module

These modules can be placed in the chassis in any order except the ICM must be in the Processor Slot, also known as a slot 17. This is the processor slot address for both the HMR and VMR 4000 Racks. In the HMR 4000 (5) Slot Rack it is the top slot. The remaining slots are 0 to 3 starting from the bottom.

For this sample application our slot assignments will be as follows:

Slot 01	ABM 4010
Slot 02	SEM 4020
Slot 06	LCM 4020
Slot 08	DSM 4010
Slot 15	IOM 4020
Slot 16	IOM 4020
P/Slot	ICM 4020

To interface the system to the Operator Control Consoles and Show Action Equipment the following definitions will be used:

Operator's Control Console - Inputs

Description	Physical Address	Channel Address	Alias
Preshow Stop PB	@r00,16,1.0	@1900.0	PRESHOW_STOP
Preshow Start PB	@r00,16,1.1	@1900.1	PRESHOW_START
Theatre Stop PB	@r00,16,1.2	@1900.2	THEATER_STOP
Theatre Start PB	@r00,16,1.3	@1900.3	THEATER_START
Preshow Entry Open PB	@r00,16,1.4	@1900.4	PS_ENTRY_OPEN
Theatre Exit Open PB	@r00,16,1.5	@1900.5	TH_EXIT_OPEN
Night Mode PB	@r00,16,1.6	@1900.6	NITE_MODE
Day Mode PB	@r00,16,1.7	@1900.7	DAY_MODE

Operator's Control Console - Outputs

Description	Physical Address	Channel Address	Alias
Preshow Stop Ind	@r00,16,0.0	@0900.0	PRESHOW_STOPPED
Preshow Start Ind	@r00,16,0.1	@0900.1	PRESHOW_RUNNING
Theatre Stop Ind	@r00,16,0.2	@0900.2	THEATER_STOPPED
Theatre Start Ind	@r00,16,0.3	@0900.3	THEATER_RUNNING
Preshow Entry Open Ind	@r00,16,0.4	@0900.4	PS_ENT_OPENING
Theatre Exit Open Ind	@r00,16,0.5	@0900.5	TH_EXIT_OPENING
Night Mode Ind	@r00,16,0.6	@0900.6	IN_NITE_MODE
Day Mode Ind	@r00,16,0.7	@0900.7	IN_DAY_MODE

Show Action Equipment - Inputs

Description	Physical Address	Channel Address	Alias
Preshow Entry Doors Closed	@r00,15,1.0	@1901.0	PS_ENT_CLOSED
Theatre Entry Doors Closed	@r00,15,1.1	@1901.1	TH_ENT_CLOSED
Theatre Exit Doors Closed	@r00,15,1.2	@1901.2	TH_EXT_CLOSED
Fire Alarm Input	@r00,15,1.7	@1901.7	FIRE_ALARM

Show Action Equipment - Outputs

Description	Physical Address	Channel Address	Alias
Preshow Entry Doors Close	@r00,15,0.0	@0901.0	PS_ENT_CLOSE
Theatre Entry Doors Close	@r00,15,0.1	@0901.1	TH_ENT_CLOSE
Theatre Exit Doors Close	@r00,15,0.2	@0901.2	TH_EXT_CLOSE
Fire Alarm Acknowledge	@r00,15,0.7	@0901.7	FIRE_ALARM_ACK

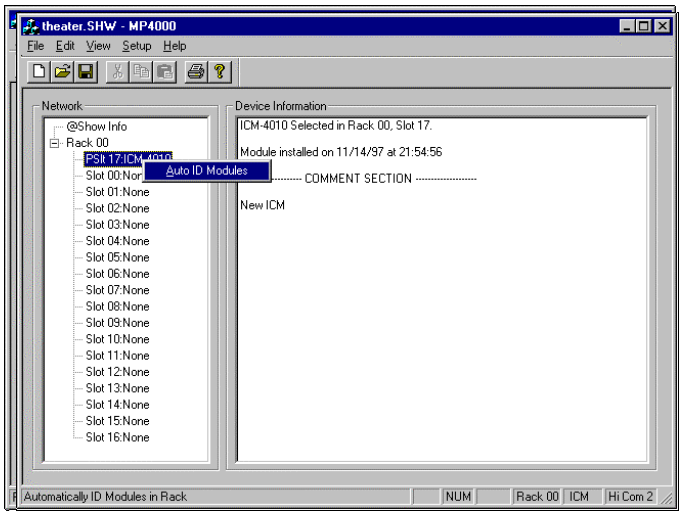
DMX Channel Definitions

Description	Physical Address	Channel Assignment
Preshow Host Key Light	@r00,6,0	LCM Channel 1
Preshow House Lights	@r00,6,0	LCM Channel 2
Theater Host Key Light	@r00,6,0	LCM Channel 3
Theater House Lights	@r00,6,0	LCM Channel 4

STEP 2 – CONFIGURE MODULE PERSONALITIES

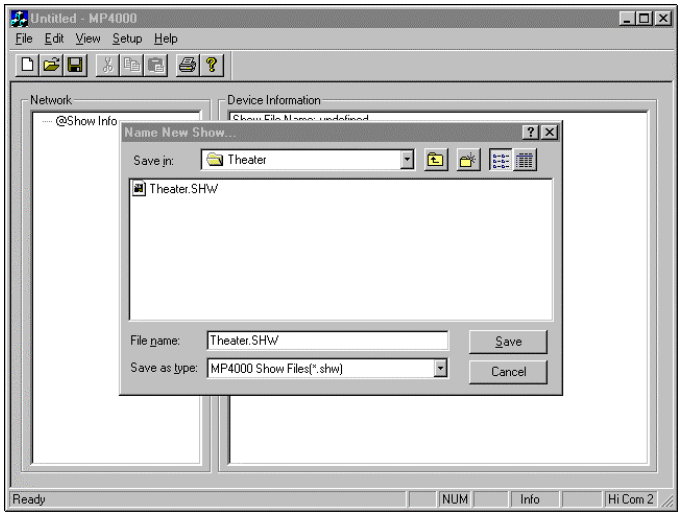
4.1.2 Theatre System Configuration

The next step is to configure the system using the MP4000 software. Double-click the ICON to start the application.

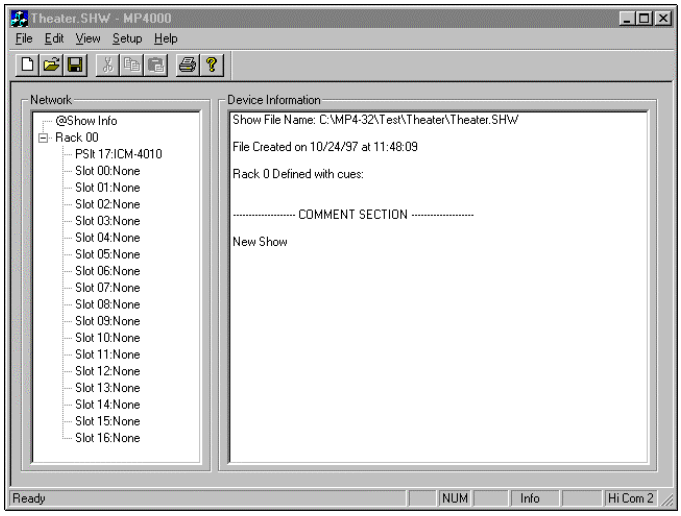


When the program begins you will see the screen at the right. The next step is to click **File** and then **New** so we can create our new show.

Pick the directory and file name for the new application. In this case we will call it Theater. All show files end with the extension .shw.



After the new show is created, the title bar displays the show name, and the network pane displays rack 0 information. We can now configure the system resources



If the rack you want to configure is available, you can have the MP4000 software automatically configure what modules are in which slot.

Left click the P Slot (17) of the rack you want to configure in the network pane to select that rack's ICM.

Right click the ICM {P Slot (17)}, a drop down list appears allowing us to automatically identify the modules in the selected module rack.

If the rack you want to configure is not available, you can configure what modules are in which slots manually.

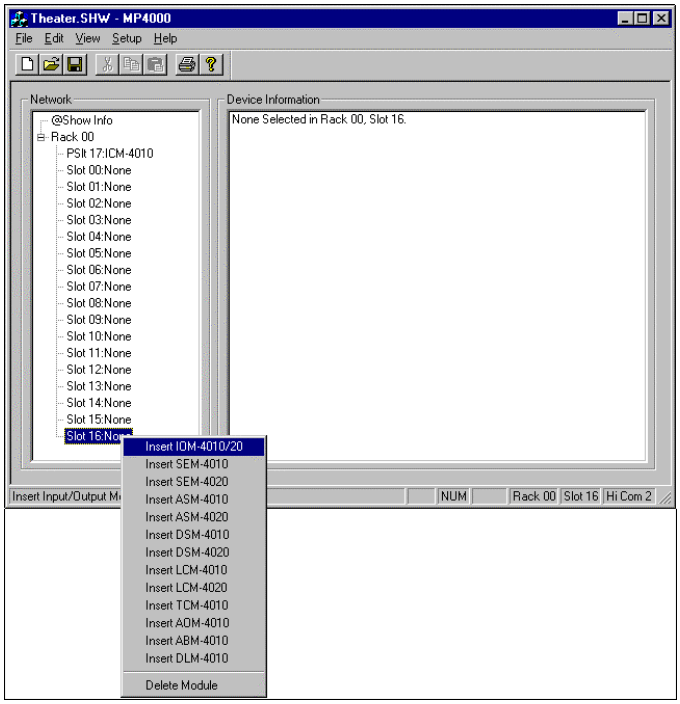
Remember, slot 17(also known as the "P" slot) is always reserved for the rack ICM. In an HMR, it is the top slot in the rack.

In a VMR, the **P-Slot** is the first slot to the right of the power supply.

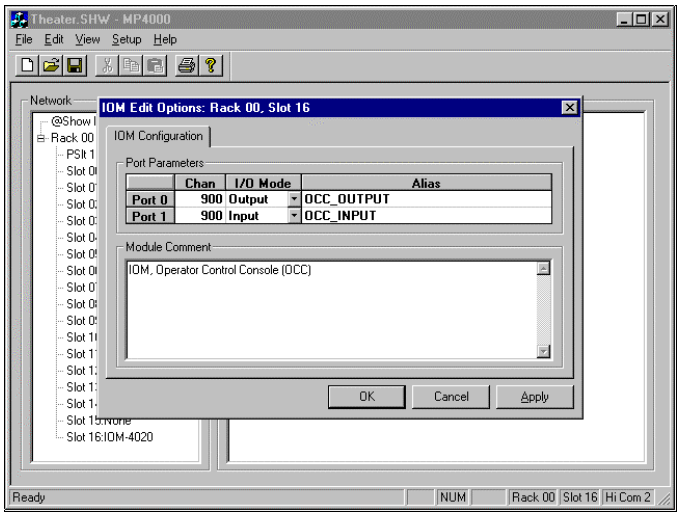
4.1.3 Theatre System IOM Configuration

By right clicking on a slot in the network pane, a drop down list appears allowing us to assign an MP-4000 module to that slot.

In this example we will assign an IOM module to slot 16.



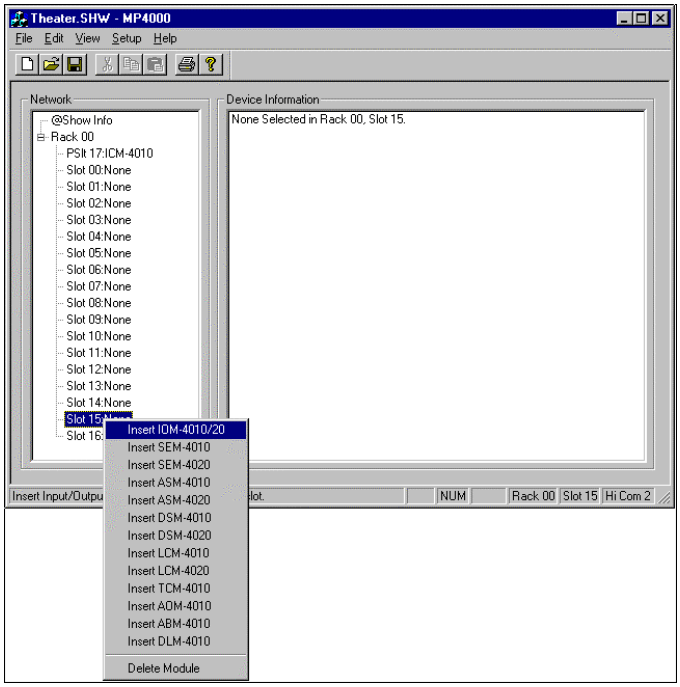
Double clicking on the IOM in slot 16 causes a set up dialog box to appear. Since this application does not require animation data to be sent to the IOM module, the Channel assignments may be above 512.



The first IOM will be used for the OCC interface. Click in the Port 0 channel box and enter 900 from the keyboard. Using the drop down list, configure Port 0 for output. Port 0 is now configured for outputs and assigned to Logical Output Channel 900. By clicking in the Alias box, we can assign an alias for this port by entering OCC_OUTPUT from the keyboard.

Using the same procedure, configure Port 1 for inputs and assign the alias OCC_INPUT. Click OK to accept the data and close the window.

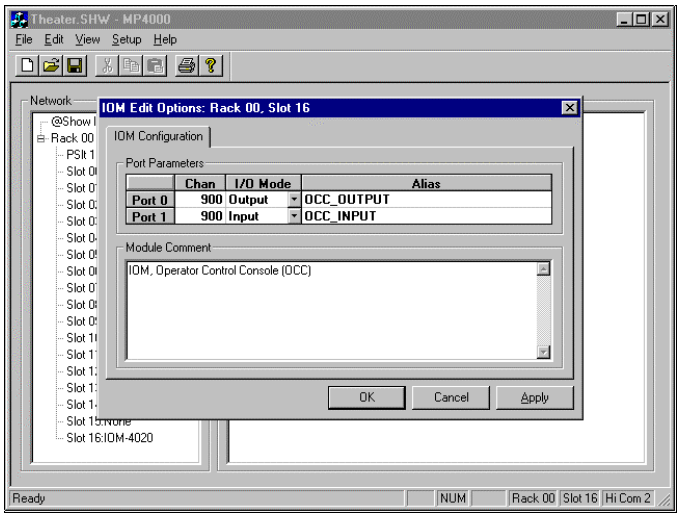
In the network window, right click on slot 15 and assign an IOM.



Double click on Slot 15 and assign Port 0 to channel 901, output mode, and assign the alias FIELD_OUTPUT.

Assign Port 1 to channel 901, input mode, and assign the alias FIELD_INPUT.

Click OK.



Tutorial

STEP 3 – DOWNLOAD RESOURCES

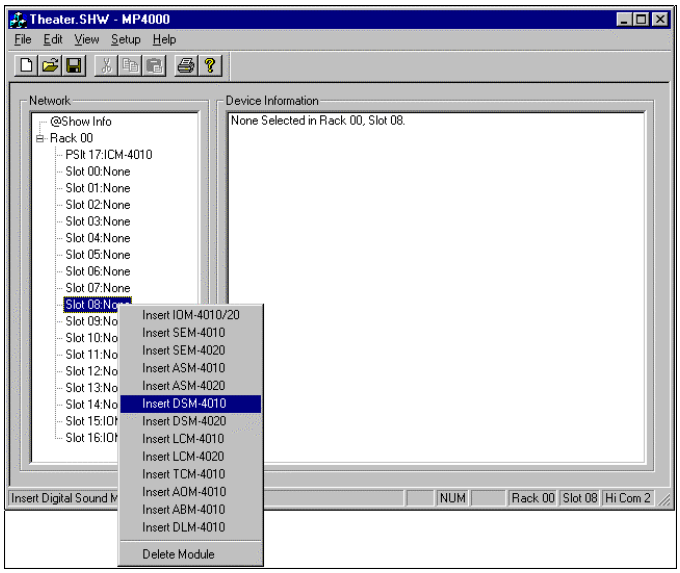
4.1.4 Theatre System DSM Configuration

In this application, DSM sound tracks will be used for Walk In/Out music.

The Walk In/Out music tracks for this application are the same but are used at different times as dictated by the venue operations. By using the ramp feature of the DSM 4000 we can control when each track is used. In addition, since the guests must transition from the entry area to the main theater, the audio must be phased together so it is continuous. To do this, the DSM is programmed in the stereo mode with both tracks containing the same music and programmed for a continuous loop. By controlling the ramp feature, the audio is independently faded up and down into the two areas.

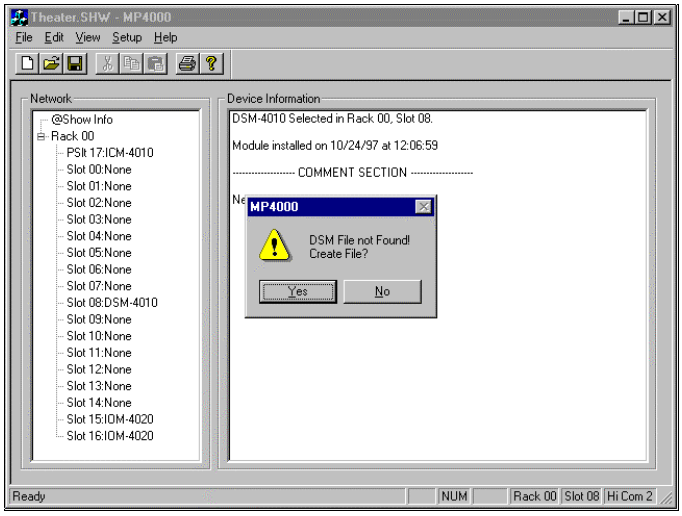
For this example we will assume that the audio track has been recorded by others and is available to us as a .WAV file on our PC. This will allow us to use the MP-4000 Software to process the sound track file and select the options we require for the venue.

Start the MP-4000 Software and open Theatre.shw. In the network pane, right click on Slot 8 and insert a DSM module.



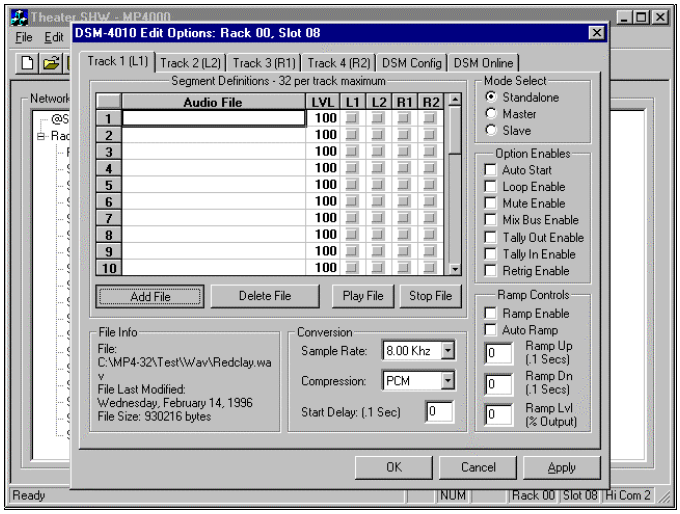
Double click on Slot 8 to configure the DSM. A message box will appear indicating that a DSM file has not been established for this module.

Click on Yes.



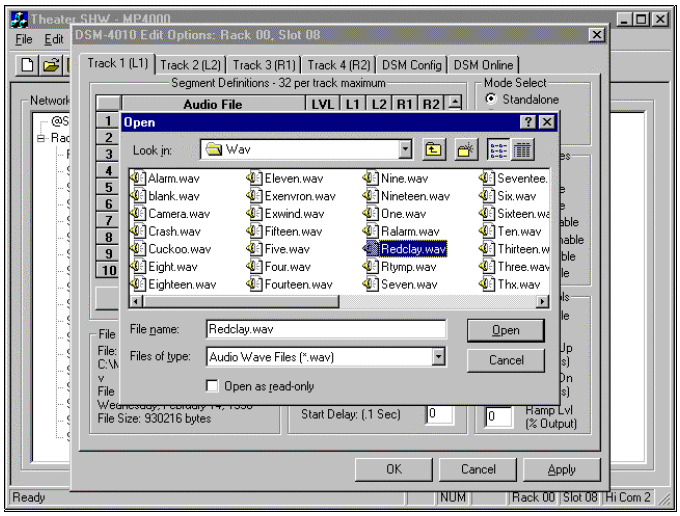
A setup dialog box appears allowing us to configure the many options on the DSM

With the Track 1 tab selected and Audio File 1 box highlighted, click Add File.



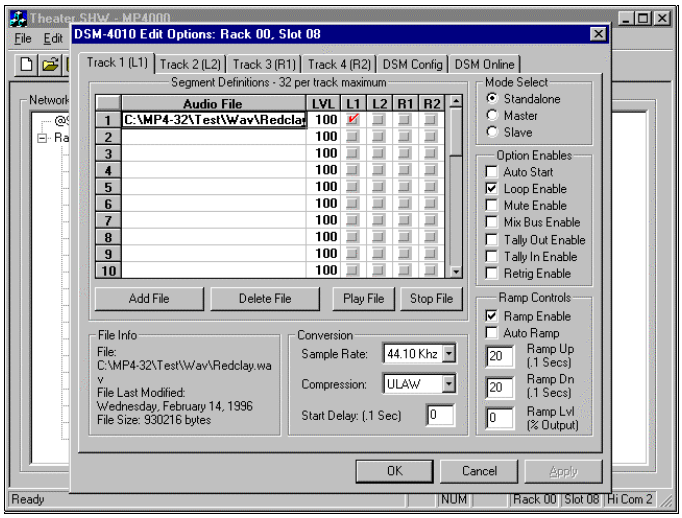
A file access dialog box will appear allowing us to select a sound file. Locate the file Redclay.wav in the Wav directory and click open.

This will assign the Redclay.wav file to play on track 1, segment 1 of the DSM module located in Slot 8.



Now that we have a sound to play, we need to tell the system where to play it. Click the check on the row our sound file is on (row 1), in the column designated L1. When the DSM is triggered the sound will play on the Left 1 output channel.

Click on the Standalone mode option button. This tell the Media Pro that this DSM works by itself and is not part of a long play chain of DSM's.

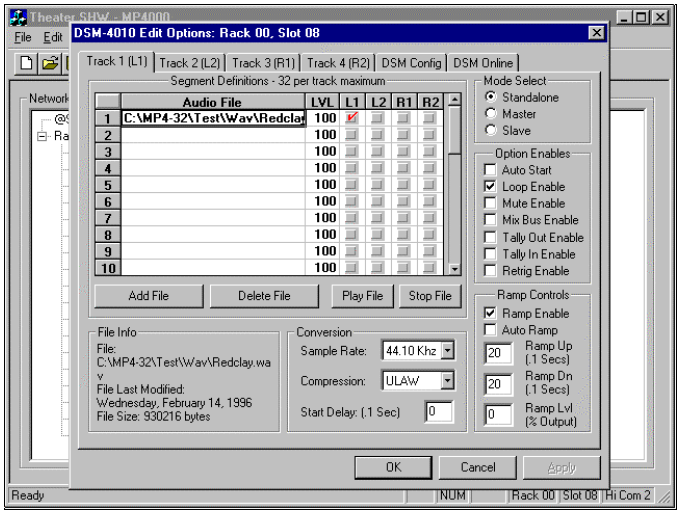


Click on the Loop Enable check box. When this DSM is triggered, it will continue to play looping the audio track until it is specifically given a stop command (programmatically or front panel stop button). When Loop Enable is not checked, the sound will stop at the end of track.

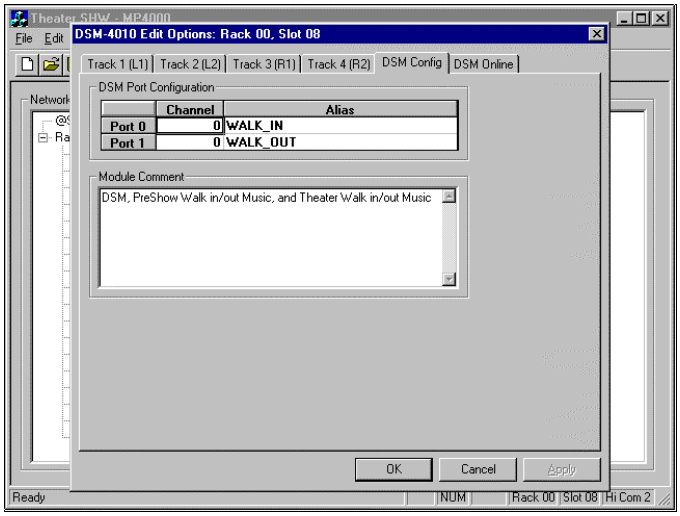
Click the Ramp Enable check box. This will allow the DSM to fade the audio in and fade the audio out when given a play command from the ICM.

Enter 20 in the Ramp Up and Ramp Down. This assigns the audio ramp rate. Values are assigned in 0.1 second increments. In this example, the audio will ramp up in 2.0 seconds and ramp out in 2.0 seconds.

Since the audio was recorded at 44.1Khz we select that option as the sample rate. We want to save memory so we also select the uLaw compression option to compress the audio in that format. We now say OK and the options are saved for this track. Track 3 is done next exactly the same way, except the R1 box is checked so that audio will go out the R1 output.



Click on the Track 3 tab. Following the above procedure, add the file Redclay.wav to segment 1 and configure the track for Standalone Mode, Loop Enable, Ramp Enable, 2 second up and down ramps, and uLaw compression. This time, click the R1 check box to route the audio out the right track.



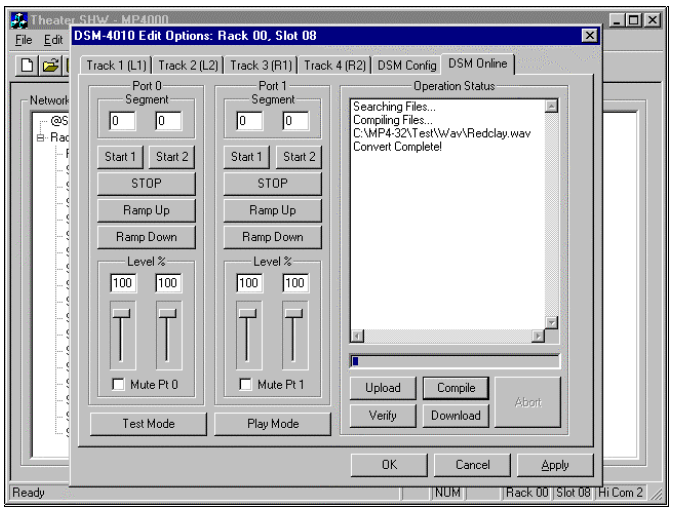
By clicking on the DSM Config tab and clicking in the appropriate text box, we can assign an alias to the two ports of the DSM.

Enter WALK_IN as the Port 0 Alias for our walk in music track.

Enter WALK_OUT as the Port 1 Alias for the walk out music track.

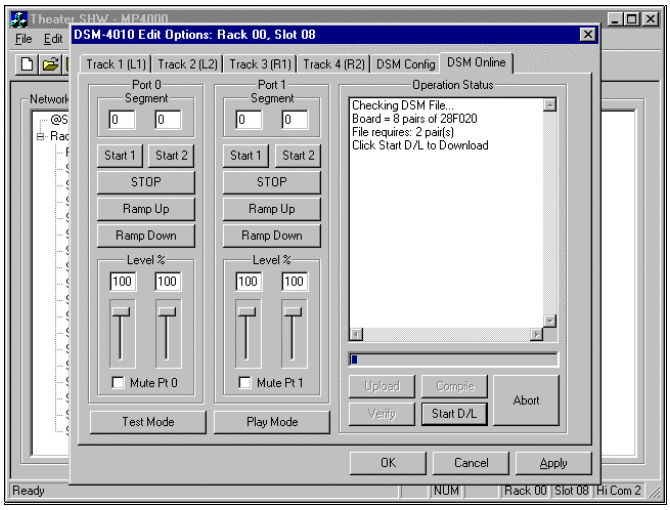
Click the Apply Button.

Now that the module options have been defined, we can go online with the DSM. Click on the DSM Online tab and click the Compile button. This will create a .DSM file that will be downloaded to the DSM using the parallel or serial connection to the Media Pro Rack.

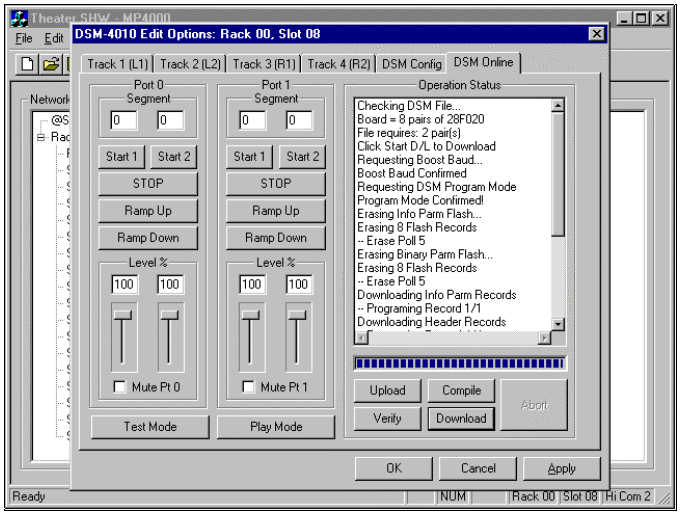


After the file has been successfully converted a message will appear in the status box indicating the memory required on board the DSM.

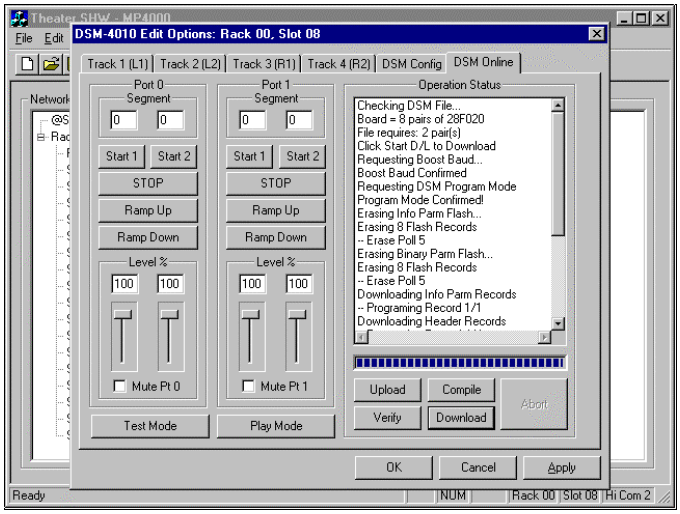
Click on the Start D/L button to begin the down load process.



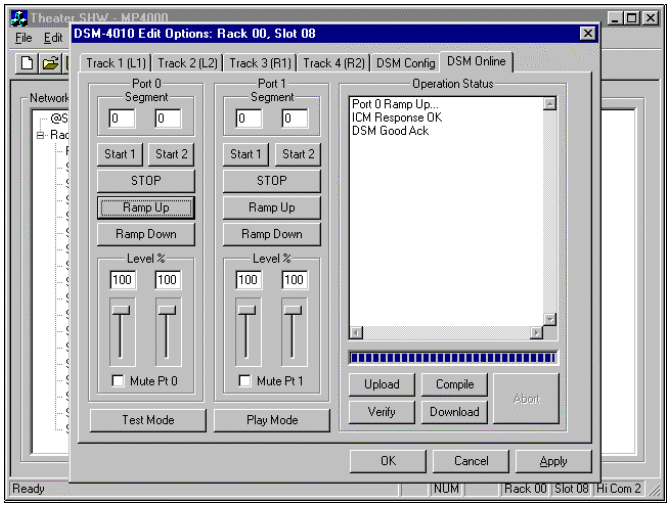
The status box now confirms the required memory is on the DSM module. Further status information is provided during the download.



After the file has been successfully transmitted to the DSM module the status box will display the download completed message.



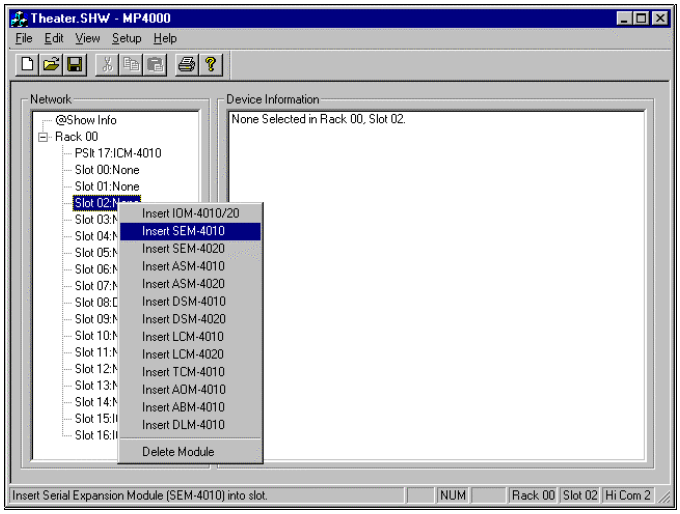
We can test that the module is working properly. By clicking on a track start button, we can verify that the track can be played. By clicking on the ramp up or down button we can verify that the audio fades properly.



4.1.5 Theatre System SEM Configuration

In our application, a Pioneer laser disk player will be used as the Pre-Show source material. We will also be using a laser disk player to simulate a film projector for the theater. A MR-16 audio matrix will be used to control the audio source material.

Start the MP-4000 Software and open Theatre.shw. In the network pane, right click on Slot 2 and insert an SEM module.

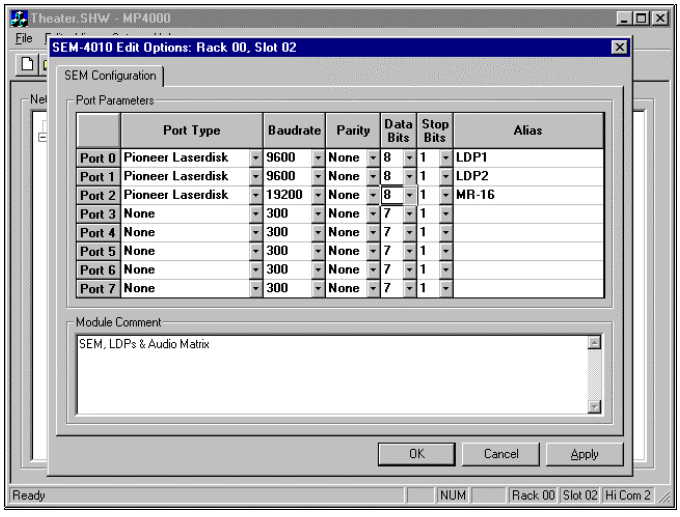


Double click on slot 2 to open the SEM configuration dialog box.

Click in the Port 0 Port Type box and use the drop down menu to select the Pioneer Laserdisk communications driver.

Click in the Baudrate box and use the drop down menu to select 9600 as the baud rate.

Using the drop down menus, select 8 data bits and 1 stop bit for the communications protocol.



Click in the Alias text box and enter LDP1 as the Port 0 alias.

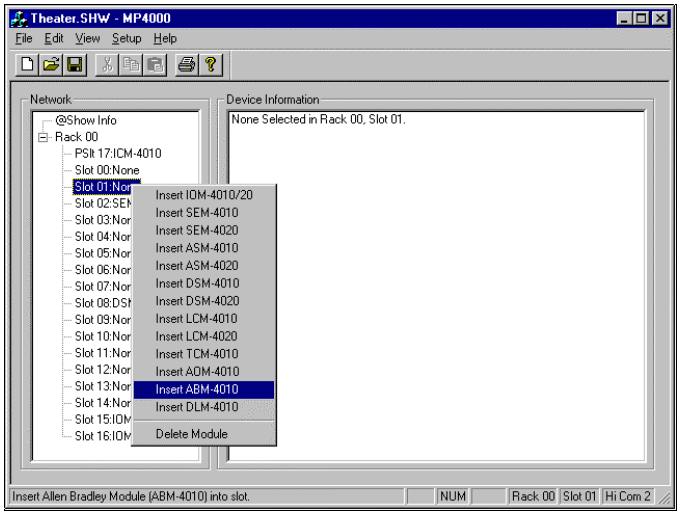
Set Port 1 to the same configuration, but this time enter LDP2 as the alias.

Configure Port 2 as a Pioneer laser disk. Set communications for 19200, no parity, 8 data bits, and no stop bit. Assign MR-16 as the alias. In this application the audio matrix will use the same communications driver as a Pioneer laser disk player. Click the OK button.

4.1.6 Theatre System ABM Configuration

We will be using an ABM to communicate with a PLC in our application. The PLC will be in control of the theater projector and will let the Media Pro know when the projector is ready.

Right click on Slot 01 in the network window and assign an ABM-4010 module.



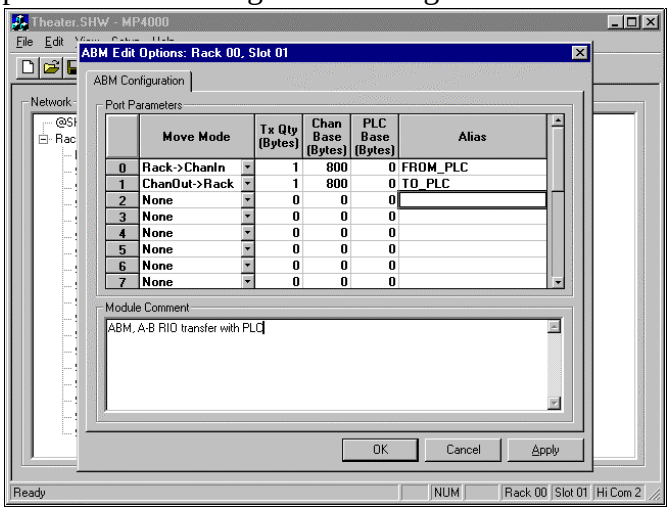
Double click on Slot 01 in the network window to open the ABM configuration dialog box.

On the Port 0 Move Mode list box select Rack->ChanIn. This will assign Port 0 to receive data from the PLC rack to the input channel data table of the Media Pro.

Because our application is small we will only need 1 byte (8 bits) of data from the PLC. Enter 1 for the Tx Qty (transmit quantity).

Enter 800 for the channel base address. This is a location we have chosen to receive the PLC data.

Click in the Alias box and enter FROM_PLC for the alias.



Set Port 1 for ChanOut->Rack. This will assign Port 1 to send data from the output channel data table of the Media Pro to the PLC Rack.

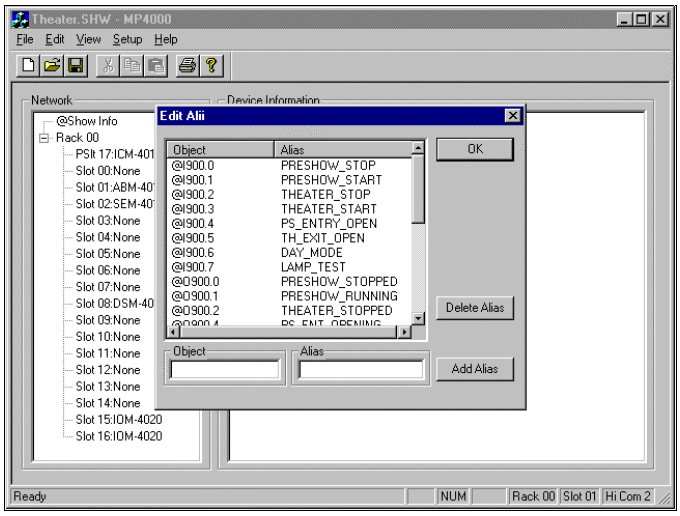
Set the Tx Qty to 1, the Channel Base address to 800, and enter TO_PLC as the alias.

Click the OK Button.

4.1.7 Theatre System I/O Alias Assignment

As previously mentioned, we can assign a simple text expression (an alias) to refer to an object in the Media Pro® 4000 system. We will use this feature in order to make our show scrip easier to read and understand, and easier to enter.

Click on the Edit menu, then click on Alii. This opens the Edit Alii window. We can assign an alias by first entering the object being referred to then entering the alias we wish to use.



When you are finished editing Alii, click the OK button.

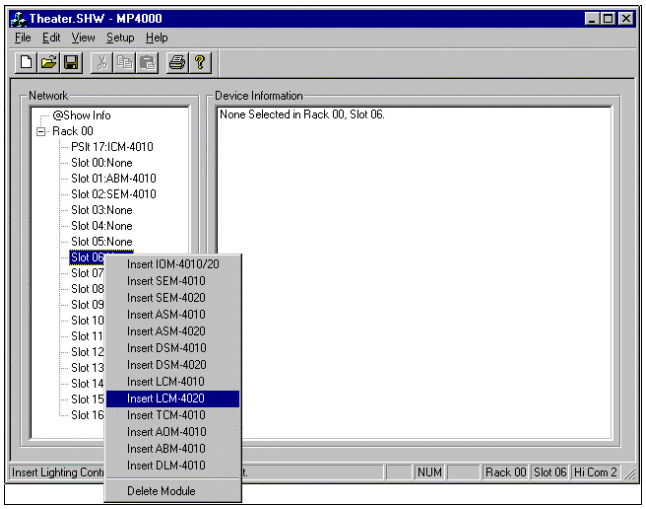
If you wish to edit an existing alias, click on the alias and it will appear in the edit window.

Here is a list of the alias assignment we will use in this application:

DAY_BLINK for @Q007
DAY_MODE for @I900.7
DAY_MODE_START for @Q002
FIELD_INPUT for @R00,15,01
FIELD_OUTPUT for @R00,15,00
FIRE_ALARM for @I901.7
FIRE_ALARM_ACK for @O901.7
FIRE_ALARM_LOOP for @Q004
FIRE_ALARM_STOP for @Q005
FROM_PLC for @R00,01,00
IN_DAY_MODE for @O900.7
IN_NITE_MODE for @O900.6
MAIN_SHW_START for @Q021
MAIN_SHW_STOP for @Q020
MP_WATCHDOG for @O800.0
MP_WATCHDOG_RUN for @I800.0
NITE_BLINK for @Q006
NITE_MODE for @I900.6
NITE_MODE_START for @Q001
OCC_INPUT for @R00,16,01
OCC_OUTPUT for @R00,16,00
PRE_SHW_START for @Q011
PRE_SHW_STOP for @Q010
PRESHOW_RUNNING for @O900.1
PRESHOW_START for @I900.1
PRESHOW_STOP for @I900.0
PRESHOW_STOPPED for @O900.0
PROJ_DAY_MODE for @O800.1
PROJ_READY for @I800.1
PROJ_RUN for @O800.2
PROJ_RUNNING for @I800.2
PROJ_START_CUE for @Q023
PROJ_STOP_CUE for @Q022
PS_BLINK for @Q012
PS_ENT_CLOSE for @O901.0
PS_ENT_CLOSED for @I901.0
PS_ENT_OPENING for @O900.4
PS_ENT_WATCH for @Q030
PS_ENTRY_OPEN for @I900.4
TH_BLINK for @Q024
TH_ENT_CLOSE for @O901.1
TH_ENT_CLOSED for @I901.1
TH_EXIT_OPEN for @I900.5
TH_EXT_CLOSE for @O901.2
TH_EXT_CLOSED for @I901.2
TH_EXT_OPENING for @O900.5
TH_EXT_WATCH for @Q031
THEATER_INIT for @Q000
THEATER_RUNNING for @O900.3
THEATER_START for @I900.3
THEATER_STOP for @I900.2
THEATER_STOPPED for @O900.2
TO_PLC for @R00,01,01
WALK_IN for @R00,08,00
WALK_OUT for @R00,08,01

4.1.8 Theatre System LCM Configuration

Right click on Slot 06 in the network window. Insert a LCM-4020 in this slot.



Double click on Slot 06 to open the configuration dialog box for the LCM.

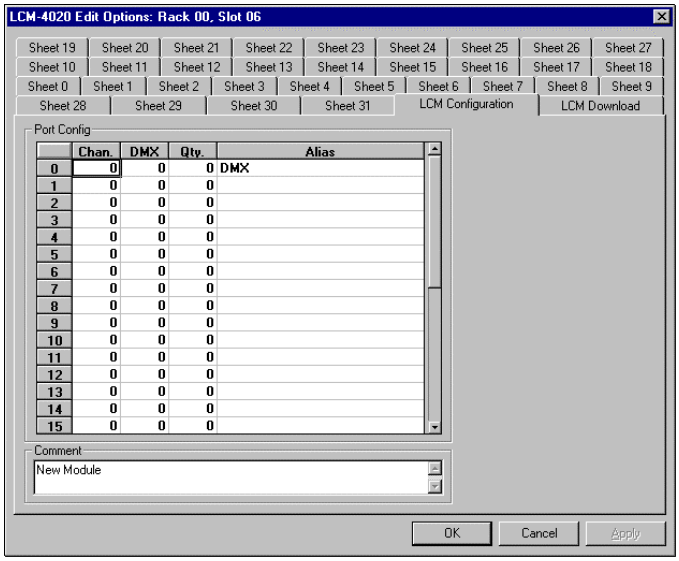
The LCM Configuration tab allows the assignment of ICM animation control channels to DMX output channels.

In this example we will not be using animation data for control.

The tabs for sheet 0 through 31 are control cue sheets for the 32 ports available on the LCM.

We will be controlling the LCM directly from the ICM in this application, so cue sheet editing is not required.

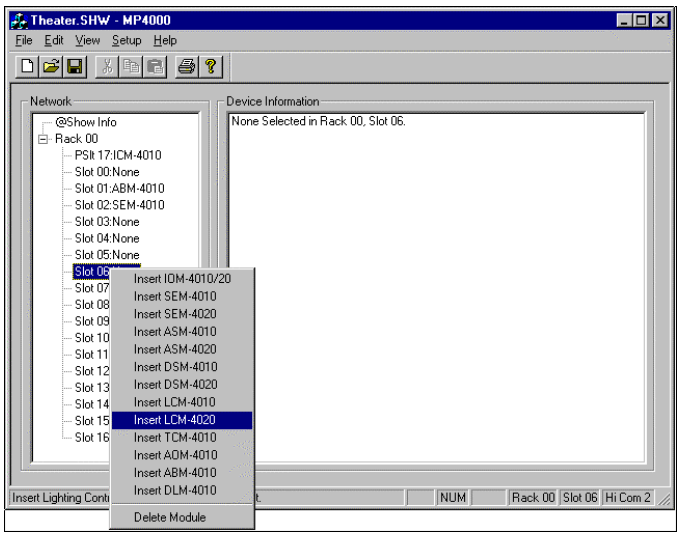
Click the OK button to close the dialog box.



4.1.9 Theatre System ICM Configuration

Double clicking on the ICM module in the network pane will open the ICM edit options dialog box.

The ICM Options dialog box opens with the ICM Cue Edit tab selected. From here cues can be scripted, edited, and can be copied and pasted.



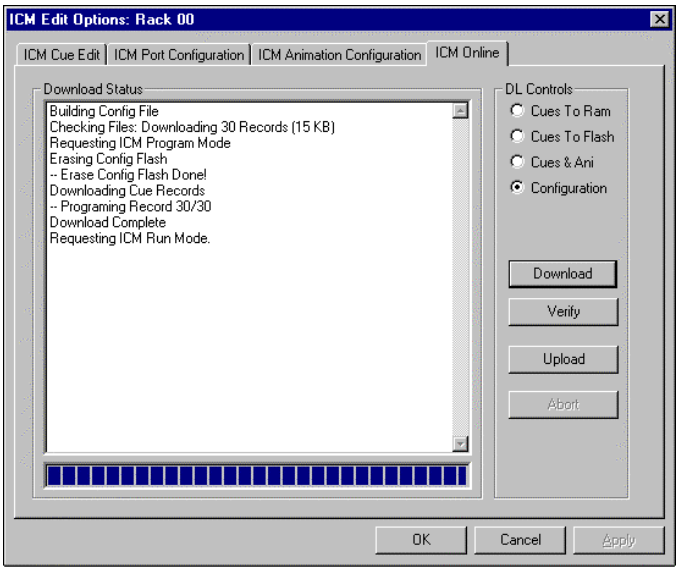
The Port Configuration tab enables the user to configure the communications parameters for the two available serial ports.

The Animation Configuration tab enables the assignment of animation data files to @a000 through @a511.

Click in the ICM Online tab, where we will download our current configuration data to the ICM.

Click on the Configuration selector button, then on the Download button. The ICM configuration file is now being built and will be downloaded to the ICM. The Download Status pane provides information regarding the progress of the download.

After the download has been completed and the ICM acknowledges run mode, click the OK button.



This completes the configuration process for this application. The next step is to script and download the cueing information needed to operate the venue.

STEP 5 - WRITE CUES

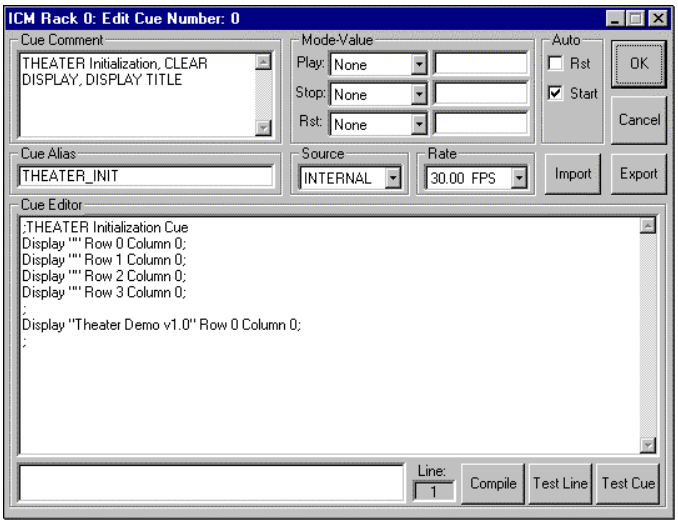
4.1.10 Theatre Show Cues

Now that the system configuration and module resources have been completed we can move on to writing the Cues for this application. Since the Cues can be organized any way the designer likes, it is arbitrary what Cue number is used for each group of events. However, it is typically accepted to establish some basic standards for applications at a given location to ease the maintenance of the software. This example will show how it has been done in the past but is only one way of doing it.

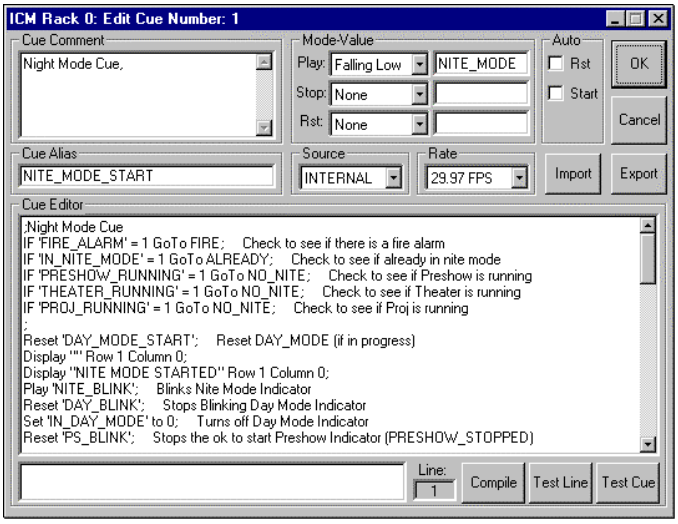
We can start by defining the Cues we need to meet the show specification. The following table lists the Cues defined for the Theater example:

@Q000	THEATER_INIT
@Q001	NITE_MODE_START
@Q002	DAY_MODE_START
@Q003	WATCHDOG
@Q004	FIRE_ALARM_LOOP
@Q005	FIRE_ALARM_STOP
@Q006	NITE_BLINK
@Q007	DAY_BLINK
@Q010	PRE_SHW_STOP
@Q011	PRE_SHW_START
@Q012	PS_BLINK
@Q020	MAIN_SHW_STOP
@Q021	MAIN_SHW_START
@Q022	PROJ_STOP_CUE
@Q023	PROJ_START_CUE
@Q024	TH_BLINK
@Q030	PS_ENT_WATCH
@Q031	TH_EXT_WATCH

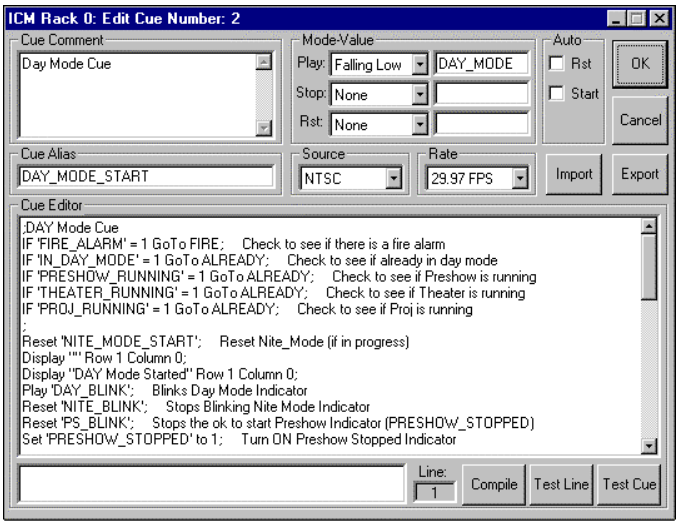
The following Cue Screens show the logic created for each Cue as described on the previous page:



CUE 00 – THEATER_INIT
Clears the HandHeld Maintenance Terminal and displays the venue sign-on message. Since the Auto Start option is checked it will execute each time the system is started or reset.

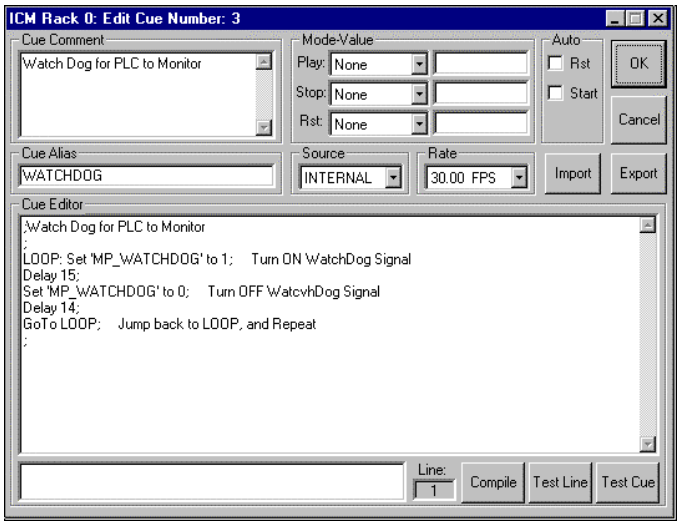


CUE 01 – NITE_MODE
This cue displays a message on the HandHeld Maintenance Terminal and then plays the stop cues to insure the shows are stopped when the Night Mode push-button is pressed. The Pre-Show and Theater music is ramped down, as is the DMX controlled lighting channels. The Pre-Show laser disk player is spun down and a message that the night mode is complete is displayed on the terminal after a 5 second delay.

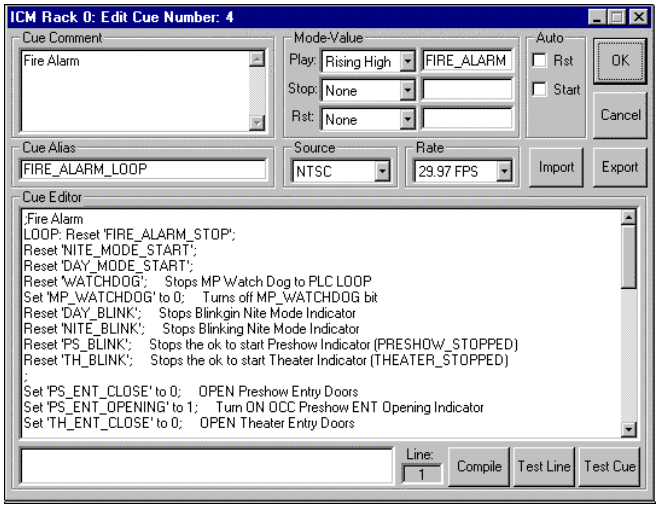


CUE 02 – DAY_MODE
This Cue puts the system in Day Mode when the Day Mode push-button is pressed. It turns off the Night Mode indicators, spins up the laser disk, plays the stop cues, starts the watchdog and turns on the Day Mode indicator on the OCC's.

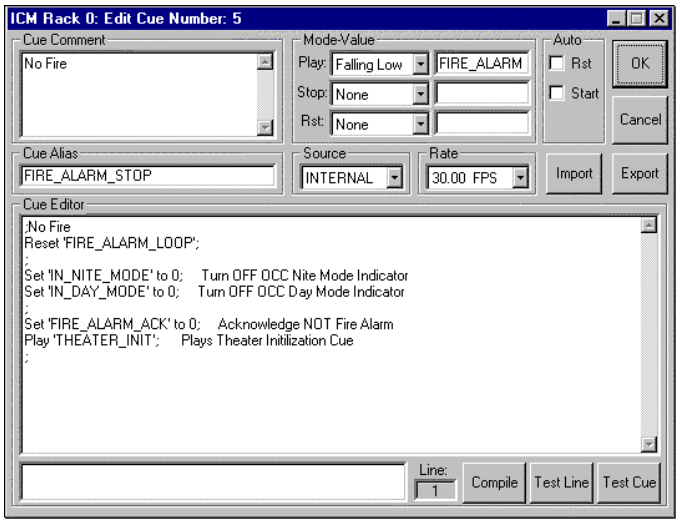
CUE 03 – WATCHDOG
This Cue cycles a bit on and off at .5 seconds intervals until stopped by playing the Night Mode Cue.



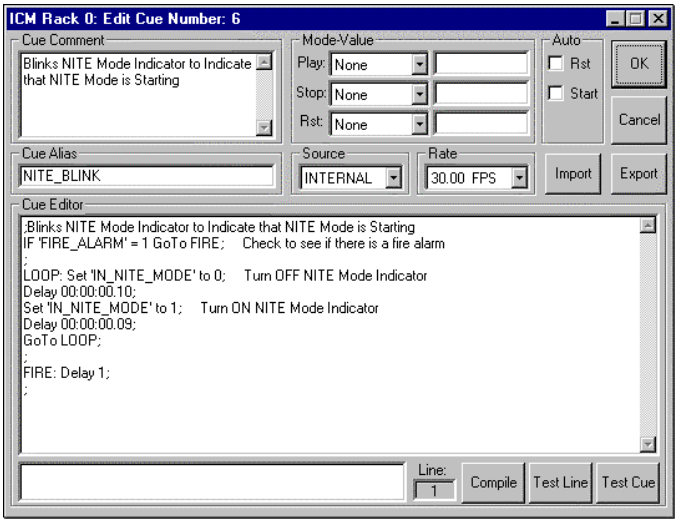
CUE 04 – FIRE_ALARM_LOOP
This cue resets the Day Mode, Night Mode, and Watchdog cues. It turns of all status indicators, opens the entry and exit doors, and turns on the door open indicators.



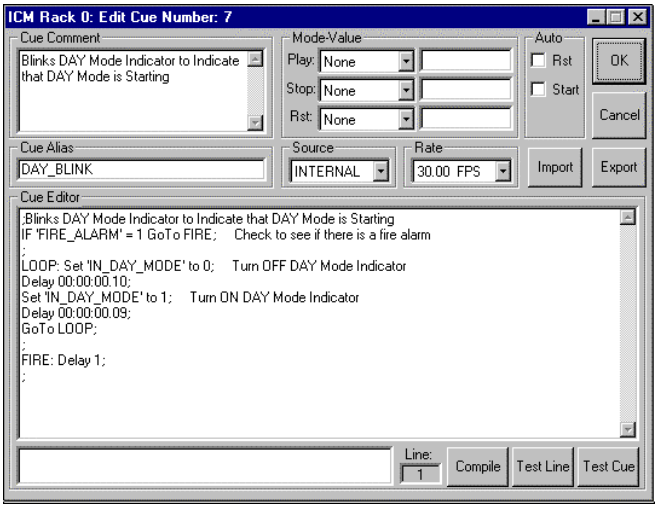
CUE – 05 FIRE_ALARM_STOP
After the Fire Alarm bit indicates there is no longer a fire alarm, the system resets back to the Theatre initialization cue.



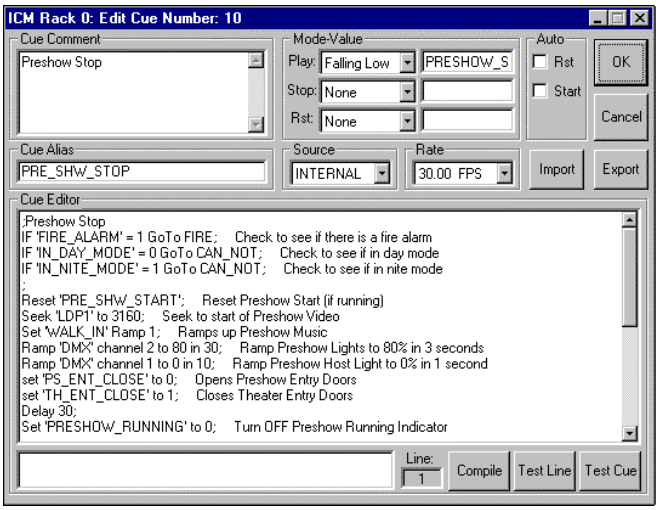
CUE 06 – NITE_BLINK
The night mode indicator light blinks while night mode is starting.



CUE 07 – DAY_BLINK
The day mode indicator blinks while day mode is starting.

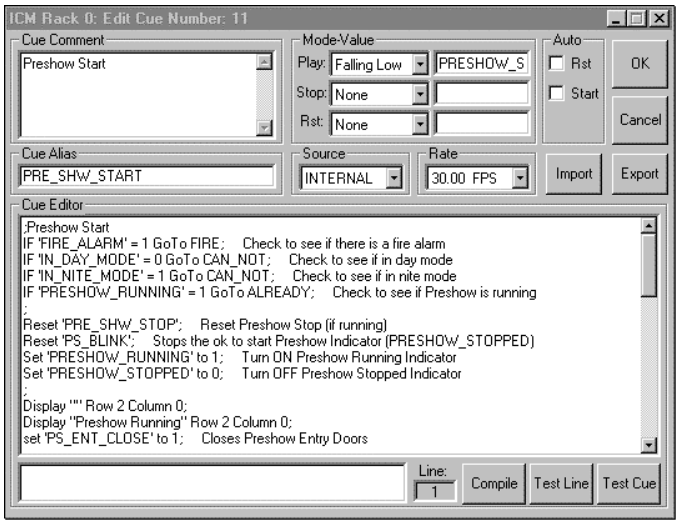


CUE 10 – PRE_SHW_STOP
This Cue resets the Pre-Show Start Cue, seeks the laser disk player to frame 3160, ramps up the Pre-Show music, ramps up the Pre-Show house lights, ramps down the Host light and turns on the OCC Pre-Show indicator.



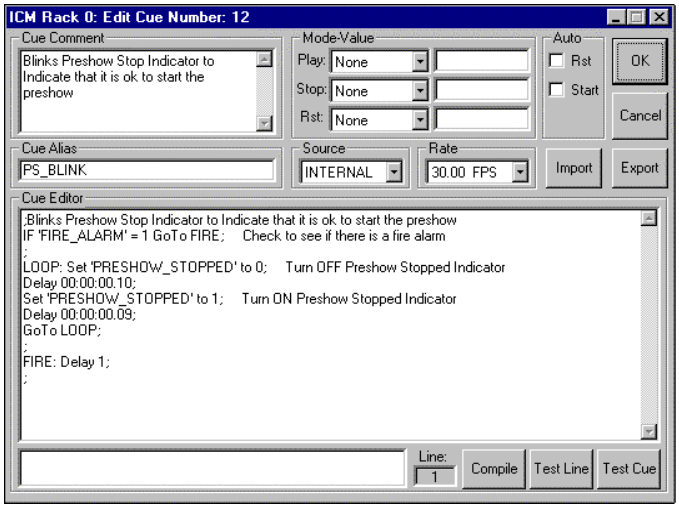
CUE 11 – PRE_SHW_START

When the Pre-Show start button is pressed, this Cue is executed, it:
changes the OCC stop and start indicators,
seeks the laser disk player,
ramps down the Pre-Show area music,
ramps up the Host key light,
ramps down the Pre-Show house lights,
waits for the Host to speak,
plays the video presentation, ramps up the house lights,
opens the Theater Entrance doors.



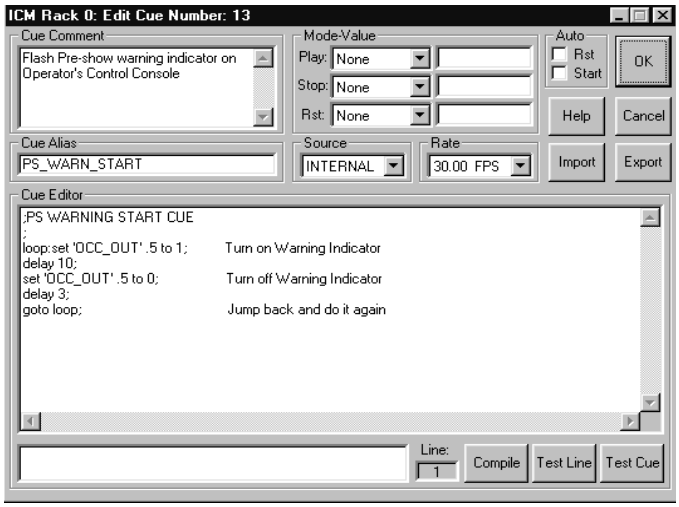
CUE 12 – PS_BLINK

This Cue blinks the Pre-Show stopped indicator light.



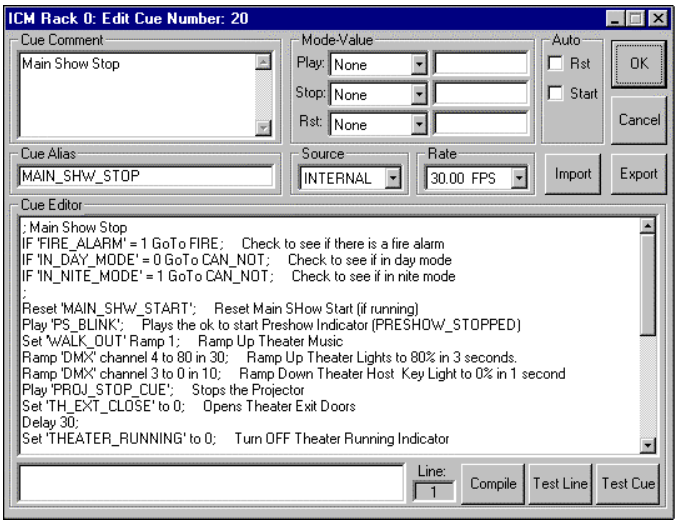
CUE 13 – PS_WARN_START

This Cue is started by the Theater Cue when it is the right time to start the Pre-Show. It flashes the Pre-Show Warning indicator on the OCC's on and off at a 10 frame rate.



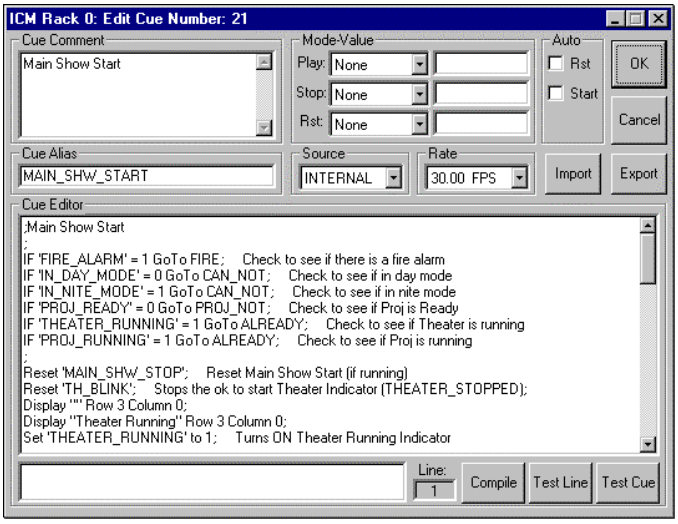
CUE 20 – MAIN_SHOW_STOP

When the Theater Show stop push-button is pressed this Cue is executed. It is also executed at the end of the Theater run Cue. It resets the Theater run Cue, changes the OCC indicator lights, ramps up the Theater music, ramps up the Theater house lights and stops the projector.



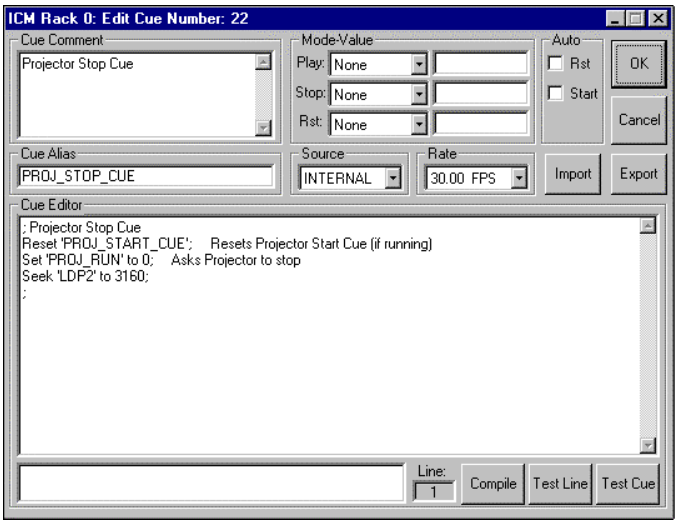
CUE 21 – MAIN_SHOW_START

This Cue is started with the OCC Theater Start push-button. It changes the state of the OCC indicators, ramps down the Theater music, ramps down the house lights, ramps up the Host key light, waits for the Host to speak, starts the projector, waits to start the Pre-Show warning Cue, waits till the end of the film, ramps up the house lights, ramps up the Theater music and opens the Theater exit doors. After the delay of the Theater emptying the music is ramped down and the doors are closed.



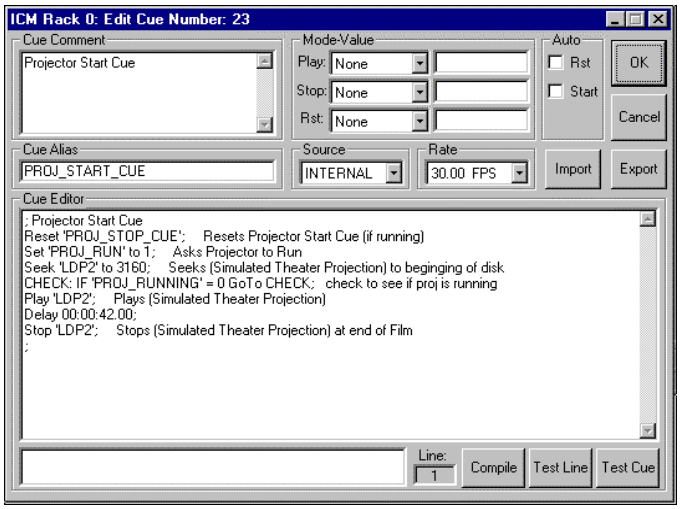
CUE 22 – PROJ_STOP

This cue is used to simulate a projector stop. The laser disk player is seeked to 3160 frames so that it is in position when we wish to start again.



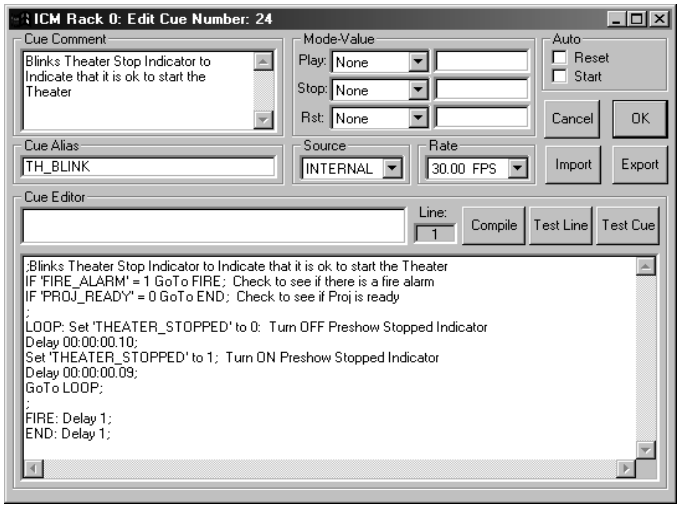
CUE 23 – PROJ_START

This cue is used to simulate a projector running. The laser disk player is seeked to the starting frame and waits for the projector run bit before starting.



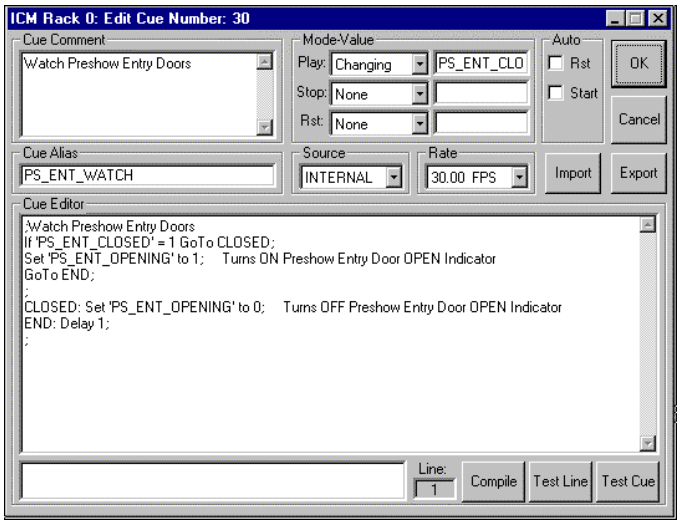
CUE 24 – TH_BLINK

This cue blinks the theater stopped indicator.

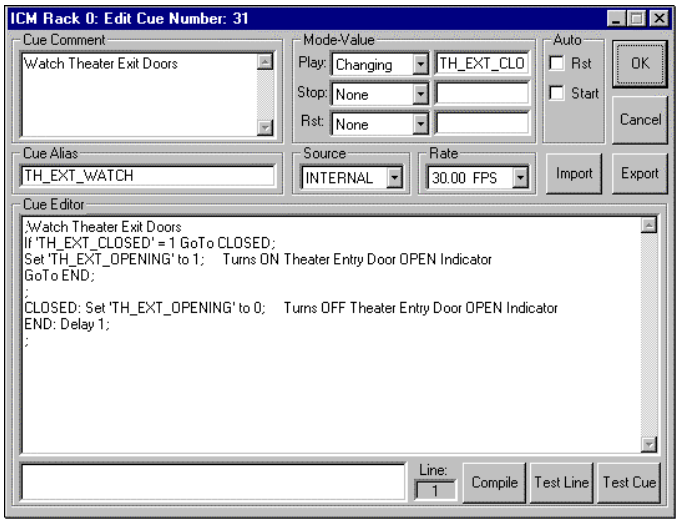


CUE 30 – PS_ENT_WATCH

This cue examines the state of the Preshow entry doors input and turns on the indicator if they are open.

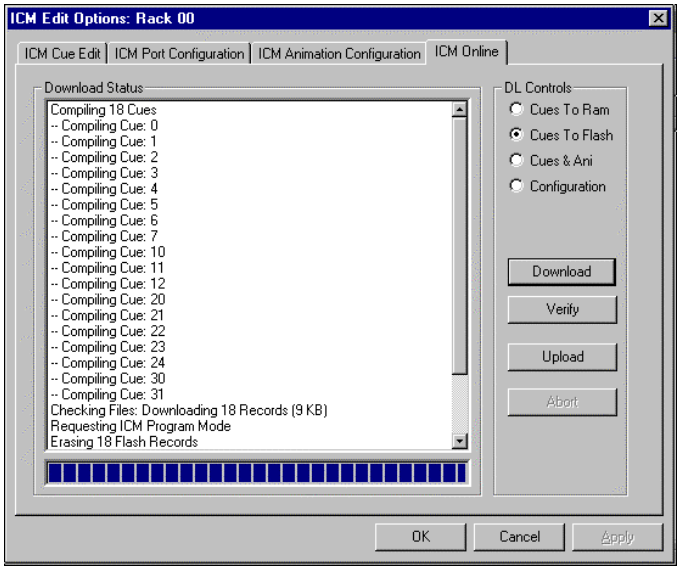


CUE 31 – TH_EXT_WATCH
This cue examines the state of the theater exit door input and turns on the indicator if they are open.

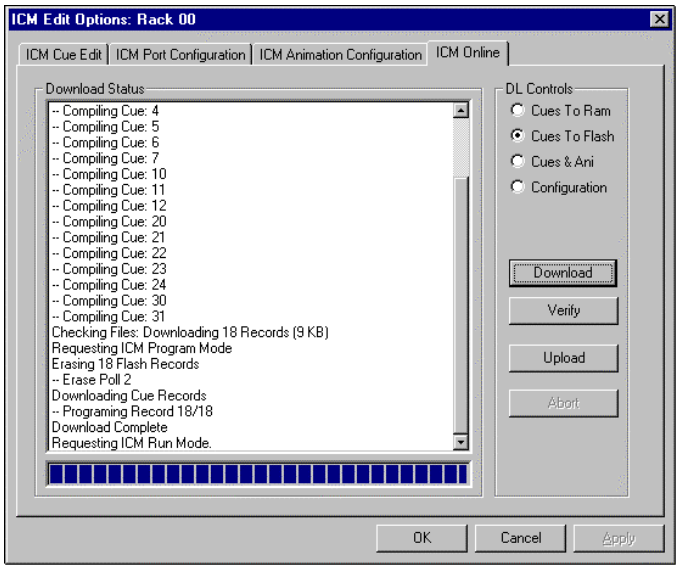


Click on the ICM in the network window. The ICM options dialog box will open. Select the ICM Online tab and click the Cues to flash option button. This will store the cue information in non-volatile memory.

Click the Download button. The Download status window will provide information regarding the download.
Once complete the Download is executed and the files are downloaded to the ICM using either a serial or parallel port connection.



Upon completion of the download, the Media Pro is reset and will start to execute the new application.
If additional testing is required the Cue edit window can be opened and single line execution can be used to test a resource connection or Cue logic. In addition, if a Cue is modified, it can be downloaded to the ICM ram for immediate execution and testing. This allows quick testing of changes before re-downloading to the flash memory. Additional testing features are available but go beyond the scope of this Tutorial.



STEP 6 – TEST THE APPLICATION

Now that the Files have been downloaded to the system, the testing phase of each cue can begin. By opening a cue sheet and selecting a given event line, single step execution of a cue can be accomplished using the Testline button. This facilitates checking connections to peripherals and verifying their proper operation.

Once each cue is proven to do the desired work, the system level interface such as an ABM or IOM module can be tested to verify the Cue is getting executed as programmed. By using the Cue Play, Stop and Reset button on the main Cue dialog, individual control of each cue is possible if the physical interface is not ready or connected yet.

This document will continue to be expanded. Please contact ASI for the latest version.

5 Section 5 – Technical Information

This section contains the ICM wiring table, Technical Application Notes and Briefs prepared as a result of frequently asked questions. The information provided is current at time of printing, however, these tech-notes are constantly being adding and updated. Please browse the ASI web site for the most recent additions:

Web http://www.Anitech-Systems.com

5.1 Connector Wiring Specifications

The following chart provides the pinout information for the ICM Field Connector wiring.

	d	b	z
2	V-Sync	V-Sync Ground	V-Sync
4	MP Network +	MP Network Shield	MP Network -
6	Port 2, Clear to Send, in	Port 2, Receive Data, in	Port 2, Ground
8	Port 2, Transmit Data, out	Port 2, Request to Send, out	Port 2, DSR / +12 / DTR
10	Port 3, Clear to Send, in	Port 3, Receive Data, in	Port 3, Ground
12	Port 3, Transmit Data, out	Port 3, Request to Send, out	Port 3, DSR / +12 / DTR
14	Port 1 {Maint}, CTS, in	Port 1 {Maint}, RD, in	Port 1 {Maint}, Gnd
16	Port 1 {Maint}, TD, out	Port 1 {Maint}, RTS, out	Port 1 {Maint}, DSR / +12 / DTR
18			
20			
22			
24			
26			
28			
30			
32			
	Rear View, Looking a ICM-4020 Male Pins		

Figure 5-1. ICM-4020 Pinout Information

5.2 Technical Application Notes

This section includes technical briefs that pertain to the Media Pro® 4000 system. These briefs here are current as of the time of writing, there may be more found on the Anitech-Systems website.

5.2.1 FileName Conventions

- Most Modules' Firmware starts with the first 3 letters as the module itself:
  - i.e.: DSM-4010 uses dsm4kd08.cod
- BOOT Firmware is indicated by **b** or **bt** in the middle of the .cod filename:
  - i.e.: DSM-4010 uses dsm41b01.cod
- Application Program (DOWNLOAD) Firmware is indicated by **D**, **dl** or **pgm** in the middle of the .cod:
  - i.e.: DSM-4010 uses dsm4kd08.cod
- The remaining ending characters in the filename indicate the version of the firmware.
  - i.e.: DSM-4010 dsm4kd08.cod

Exception: The ICM download (application program) Firmware is named as follows:

- First 2 characters reflect the MP4000 software version
  - **16** = 16 bit software **32** = 32 bit software.
- The next 2 characters reflect the ICM Module version
  - **10** = ICM-4010 **20** = ICM-4020
- The remaining ending characters in the filename indicate the version of the firmware.
  - i.e.: 3220109b.cod is for **32**bit MP4000 Software, ICM-40**20**, Firmware version **109b**.

5.2.2 PCMCIA Device Partitioning and Formatting

There is more than one way to partition and format a PCMCIA ATA device. Only one example is given here. Partitioning and formatting should be performed by personnel technically competent with PC/Windows. Incorrect actions can result in a complete loss of data on the PC's Hard Drives and/or result in a completely inoperable PC.

☑ The PC must have a properly configured, and operating PCMCIA Socket.
☑ The Socket *must* be Supported in the Windows Device Manager.

❖ Preparing to Partition or Format

- Insert a PCMCIA ATA Flash Drive, or PCMCIA ATA Hard Disk into a PCMCIA socket on the PC.
- In Windows 95/98 click on Start Menu, Settings, Control Panel, System, Device Manager, Disk Drives, Generic IDE Disk Type 80.
  - Note original settings before making any changes.
- Settings Tab
  - Int 13 unit will need to be selected in order to partition the device with FDISK.
  - Removable should also be selected.
- General Tab, check the Device Status, should say "This device is working properly".
- Windows may require restarting after making changes before the changes are in effect.

❖ Partitioning

- In Windows 95/98 click on Start Menu, Run on the open line type: **FDISK** ; click OK.
- If asked "do you wish to enable large disk support", select **No**.
- Pick option "**5**. Change current fixed disk drive".
- Select the disk that corresponds to your PCMCIA device.
  - ! DO NOT SELECT DISK 1, DRIVE C !
- At the top of the window it will show which fixed disk is currently selected.
  - ! If **"Current fixed disk drive: 1"** is displayed, Hit the ESC key to exit FDISK and consult PC/Windows technically competent personnel !
  - **! Proceeding any further will likely result in permanently wiping out all data the hard drive and making your PC completely unusable** !
- Select option "**4**. Display partition information"
  - If there is an existing partition listed, it will have to be removed in order to create a new partition.
  - An existing partition might look like:

Partition	Type	Mbytes	System	Usage
D: 1	PRI DOS	210	FAT16	100%

- If an existing partition exists: Select option "**3**. Delete partition or Logical Drive".
  - Choose the correct option based on the previously displayed partition information.
    - "1. Delete primary DOS Partition"
    - "2. Delete Extended DOS Partition"
    - "3. Delete Logical DOS Drives in the extended DOS Partition"
    - "4. Delete Non-DOS Partition"
  - If there were Logical DOS drives in an extended DOS partition, they will need to be deleted before the Extended DOS partition can be removed.

- If there are no existing partitions:
  - Select option "1. Create DOS partition or Logical Drive".
    - Select option "1. Create Primary DOS partition".
      - When asked "Do you wish to use the maximum available size for a primary DOS partition", select **Yes**.
- After the partition is created, Press the **ESC** key to continue.
- Press the ESC key to Exit FDISK.
- A Message will probably appear:

"You MUST restart your system for your changes to take effect.
Any Drives you have created or changed must be formatted AFTER you restart.
Shut down Windows before restarting."

- Press the ESC key to Exit FDISK.
- In Windows 95/98 click on Start Menu, Shut Down
- Shut Down the Computer, Yes

Congratulations, you have either successfully partitioned the PCMCIA ATA Flash Drive, or PCMCIA ATA Hard Disk in your PCMCIA socket, or wiped out all data on a hard drive.

Proceed to the next step, Formatting, on the following page.

Formatting

- In Windows 95/98 click on Start Menu, Programs, MS-DOS Prompt.
- At the DOS prompt type: **Format x /q**
  - The /q stands for Quick Format.
  - Where **X** is the Drive letter assigned to your PCMCIA ATA Flash Drive, or PCMCIA ATA Hard Disk.
  - **! DO NOT FORMAT DRIVE C !**
- If you can not figure out the drive letter for the PCMCIA ATA Flash Drive, or PCMCIA ATA Hard Disk.
  - Hit the ESC key to clear the command line.
  - Type EXIT on the command line and hit return, to exit to windows.
  - ! Consult PC/Windows technically competent personnel !
  - **! Proceeding any further will likely result in permanently wiping out all data the hard drive and making your PC completely unusable** !
- A Message will probably appear:
- "WARNING, ALL DATA ON DISK DRIVE X: WILL be LOST!
- Proceed with Format (Y/N)"
- If the drive letter is C: Hit **N**o.

❖ Type EXIT on the command line and hit return, to exit to windows.

- ! Consult PC/Windows technically competent personnel !
  - ! Proceeding any further will likely result in permanently wiping out all data the hard drive and making your PC completely unusable !
- If the drive letter **is** the drive letter of the PCMCIA ATA Flash Drive, or PCMCIA ATA Hard Disk: Hit **Y**es.
  - If a Message appears "Invalid existing format. This disk cannot be QuickFormatted. Proceed with unconditional Format (Y/N)?"
    - Select **Y**es

❖ A Message will be displayed while the drive is formatted -
"Formatting ###.##M, ###% percent completed."

- When Prompted for a Volume Label, Press Enter Only.
- Type **EXIT** on the command line and Press return, to exit to windows.
- In Windows 95/98 click on Start Menu, Shut Down, Shut Down the Computer, Yes

Congratulations, you have either successfully formatted the PCMCIA ATA Flash Drive, or PCMCIA ATA Hard Disk in your PCMCIA socket, or wiped all data on a hard drive.

❖ Compatible with DOS File Allocation Table 16 bit "Microsoft" type (DOS FAT 16) –

- Typically drives smaller than 16MB will use 12 bit DOS FAT, or can not be made to use 16 bit DOS FAT.
  - DOS FAT 12 should be avoided due to compatibility problems.

- Some drives larger than 850MB will use 32 bit DOS FAT, or can not be made to use 16 bit DOS FAT.
  - DOS FAT 32 should be avoided due to compatibility problems.

- Typically drives larger than 4GB will require 32 bit DOS FAT, and can not be made to use 16 bit DOS FAT.

    - DOS FAT 32 should be avoided due to compatibility problems.

The PCMCIA ATA Flash Drive, or PCMCIA ATA Hard Disk should now be ready for use.

6 Section 6 – Appendix

The following pages contain technical briefs relating to this module. There are additional technical briefs on the ASI website and new briefs are added on an on-going basis. Please visit the website for updated information –

http://www.Anitech-Systems.com.

Updated manuals are placed on the Web periodically. Please check to see if a more recent revision is available on the website. Revision numbers are located in the footer of the manual pages.

Also, a glossary of terms is in this section, for terms used by ASI as well as terms used in the Show Control Industry.

6.1 Saving Files from a Web Page without Displaying -

This is useful for retrieving a non-formatted page that is not intended for viewing:

From Internet Explorer® Browser

1 - Saving files from a Web page.
  A) To save a file without opening it:
    1) **Right**-click on the link for the item you want, a pop-up menu appears,
      (a) Select, then Left-click **Save Target As**
      (b) Enter the desired folder and filenames and select **Save**

From Netscape® Communicator Browser

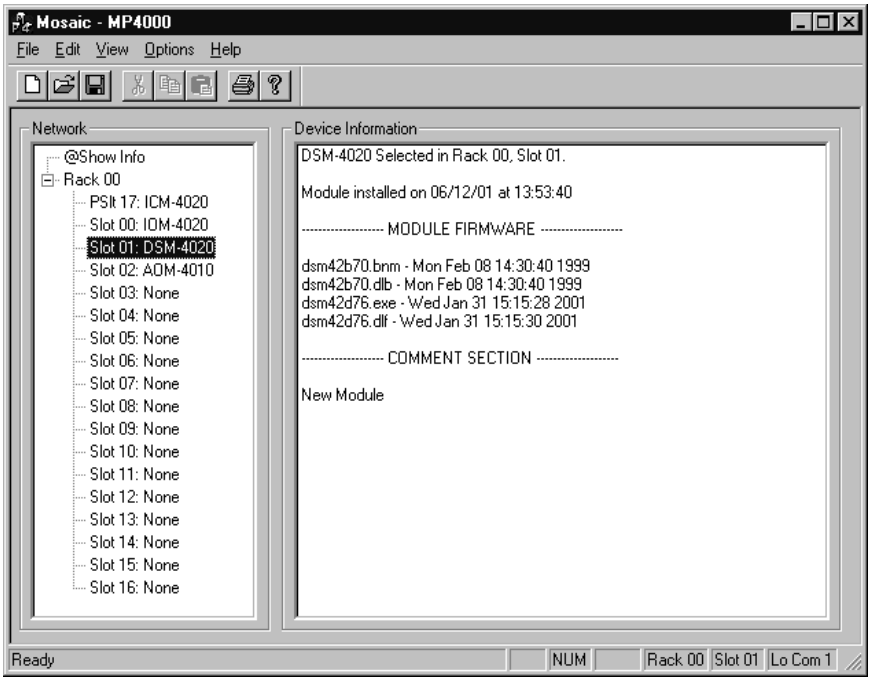
1 - Saving files without displaying them.
  A) To save a file without viewing it:
    1) **Right**-click on the file's link to display a pop-up menu,
      (a) Select, then Left-click **Save Link As**
      (b) Enter the desired folder and filename and select **Save**

6.2 Downloading New Version of Firmware to a Module

There may be circumstances where the firmware version of a module may need to be upgraded in the field. The information in this section will describe the process of downloading a new version of firmware from the Anitech Systems website and installing it into the module.

Verify Current Version –

- ✓ Open the MP4000 software.
- ✓ Click on the desired module.
- ✓ Module firmware versions are displayed in the Device Information section of the screen.



Download Firmware Code From Website –

- ❖ Visit the Anitech Systems website (www.anitech-systems.com).
  - ➢ Click on the Support link.
  - ➢ Scroll to the correct module type.
  - ➢ Highlight and select the Firmware column.
    - ▪ Select the correct module version.
      - Select the appropriate sub-file, this is dependant on the module revision level. (Refer to File Name Conventions in the Media Pro® 4000 manual, section 5.2.1, for a detailed explanation of firmware file names.)
      - When the file is selected, it needs to be saved to the disk. Since browsers vary, refer to the brief, "Saving Files from a WebPage without Displaying " in Section 6.1 of this manual.
        - ♦ Enter the Directory and Filename information on the pop-up Menu and press the Save button. The file will be saved to the disk.
- ❖ Also, carefully read the most recent ReadMeMP.txt. It is found on the website at the following address:

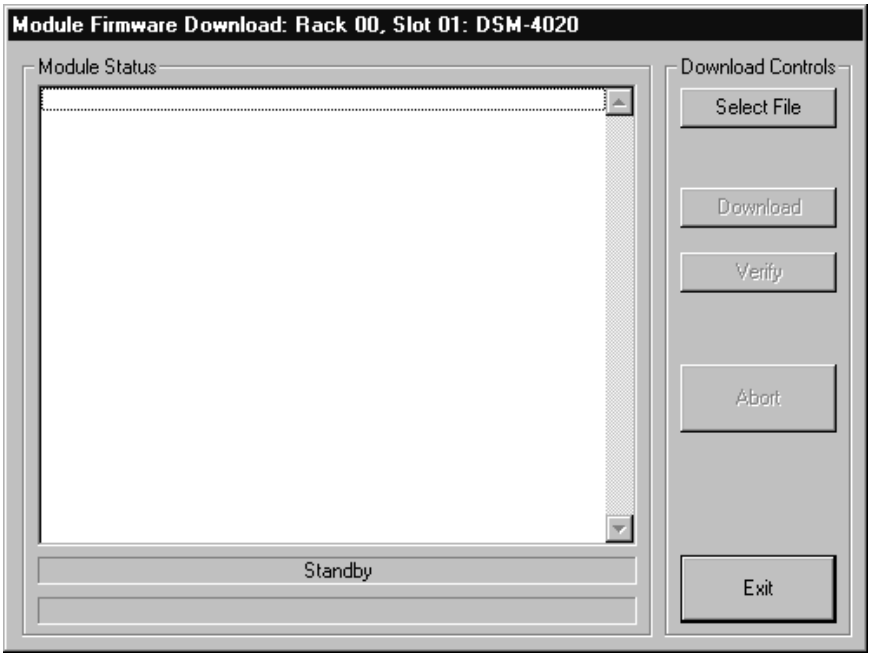
http://anitech-systems.com/MP4000/manual/briefs/ReadMeMP.txt

  - ➢ Exit the Website.

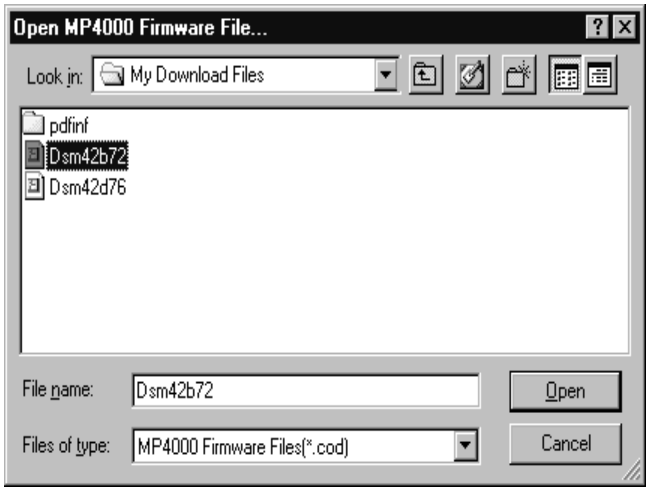
Return to the MP4000 Software –

- **Note:** Before attempting to download the firmware to a module, place the switch on the ICM front panel to the **Test** position. This will enable the rack to download faster and with minimal interference. Placing the switch in the test position affects the racks as described below:
  - Rack 0 (master) – Allows the Rack to concentrate on the download information. Quits the cues and animation from running and stops the broadcast over the MP Net.
  - Other Racks (slaves) – Still receives the information, but ignores it while the download is being performed.
  - **NOTE:** When downloading to a slave rack over the MPNet, turn the Rack 0 ICM switch to the Test position in order to minimize interference and enable a successful download.

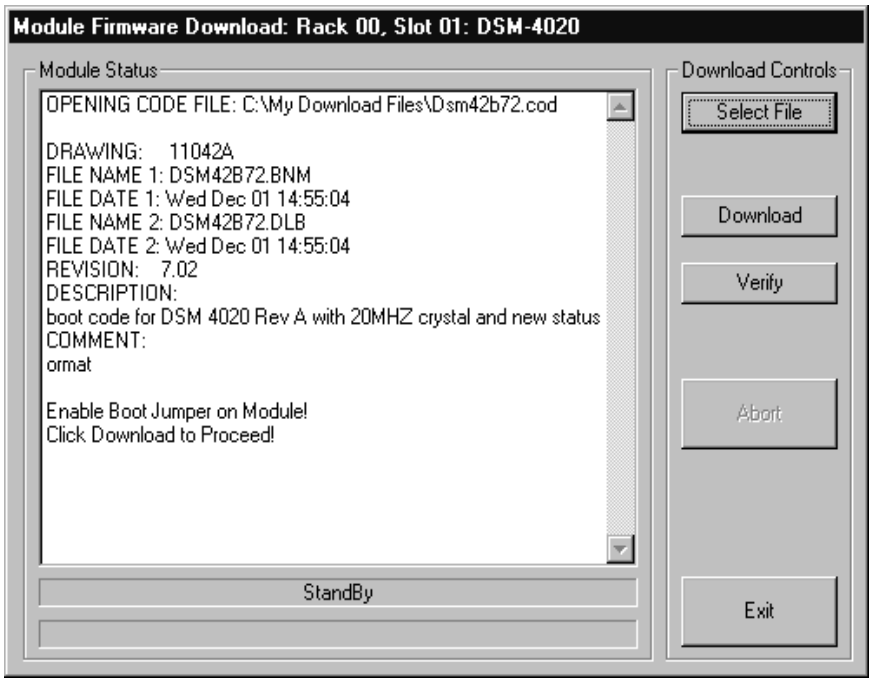
- Open the MP4000 software and select the Showfile.
- Highlight the module desired and left click. The module's information is displayed in the Device Information section.

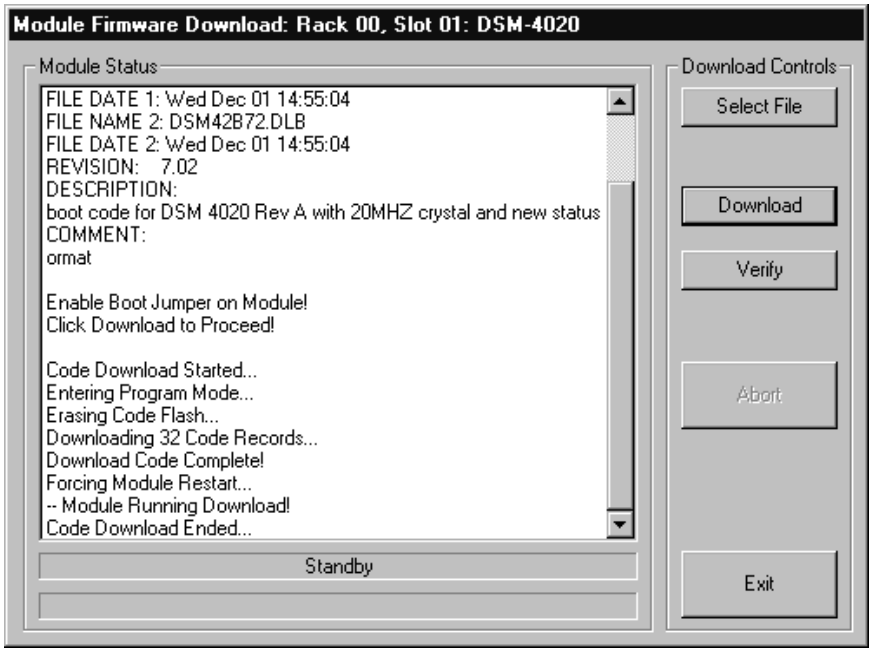


- Select OPTIONS from the Menu bar across the top of the MP4000 Menu screen.
  - From the scroll down menu, highlight and select **MP4000 FIRMWARE**. The Download menu appears.
    - In the Download Controls section of the menu, press the **Select File** button. A menu pops up for the selection of the file. Find the directory and filename, then double-click to select and open it.



Install the firmware Code in the Module –





- The download screen reappears and displays the firmware information in the Module Status section of the screen.
- Be Certain to Read the Description & Comment sections of the firmware information for considerations and restrictions of use.
- Refer to the ReadMeMP file found on the Website.
- Press the Download button to proceed, or Exit to abort the operation.
  - Once the Download has started, messages appear on the screen to detail the process that is taking place. When it is finished, **Code Download Ended** message is displayed.
- It is recommended to run a verify after the download on modules that have the option available. Press the **Verify** button and watch for the **Code Verify Ended** Message.
  - If another download is necessary, then press the Select File button and repeat the download process for the next file needed.
  - Press the **Exit** button to end the download process.

The new versions of the firmware will be displayed in the Device Information Pane of the MP4000 menu.

Glossary

ALIAS English representation contained within single quotes that references an object.

.ani Extension used for Animation files.

ASCII A numeric code used by computers to represent characters.

BIT A single logical or physical resource that can be either ON (true, 1) or OFF (false, 0).

COMMENT Text following a semicolon on an event line within a cue, it is used to annotate the cue.

COMPLIANCE Additional feedback applied according to the equations in the firmware and parameters supplied by the user that tends to reduce the output signal.

CONST Constant

CONTACT CLOSURE Completes a circuit. A voltage is applied to a reference pin of a bit's connector. When the switch is closed, the voltage is returned on another wire to the same bit, completing the circuit. When the switch is closed, the I/O pin gets voltage.

CUE A list of up to 512 EVENTS containing verb, object, token, and variable items.

db Decibel

E/S Emergency Stop

EVENT A single line of verb, object, token and parameter items contained in a CUE.

FET Field Effect Transistor - an electronically controlled switch.

FLASH Fast, electrically erasable, and programmable in the circuit non-volatile memory devices.

FW Firmware

HMR Horizontal Module Rack

HW Hardware

ICM	Intelligent Controller Module
I/O	Input/Output
IOM	Input/Output Module
JMP	Push-on jumper, shunt - a small (approximately ¼" x 1/8" thick) piece of plastic with a metal insert. The jumper is set by pushing it down over a pair of pins.
KEY	A piece of hardware that keeps a module from being plugged into a slot that is wired and keyed for a different kind of module.
LABEL	A name, followed by a colon on a line of a CUE, used for a forward or backward jump.
LED	Light Emitting Diode (indicator, light, lamp)
MIDI	Musical Instrument Digital Interface
MPCL	Media Pro® Control Language
OBJECT	Term used to reference a resource within the Media Pro® system.
OFF	False, 0, Open
ON	True, 1, Closed
PARAMETER	The item that follows a token. The list of parameters is in section 3 of the MP4000 User's Operation Manual.
PLC	Programmable Logic Controller
PORT	A resource on a module in case of the DSM. There are 2 ports.
P-P	Peak to Peak
RACK	A card cage containing Media Pro® 4000 Modules.
RESOURCE	A controllable device connected to or contained within the Media Pro® 4000 system.
RMS	Root Mean Squared
SLOT	A physical location in a RACK that contains a Media Pro® Module.

SMPTE Refers to **S**ociety of **M**otion **P**icture & **T**elevision **E**ngineers, and the standard for timing signals set by this group.

STRING A collection of alphanumeric characters contained in double quotes, used with the DISPLAY command.

SW Software

THD & N Total harmonic distortion and noise

TOKEN Reserved words for modifiers of the VERBs used in a cue event. The list of TOKENs is in section 3 of the MP4000 User's Operation Manual.

TYPE The kind of OBJECT being referred to in a direct addressing statement. The list of TYPEs is in section 3 of the MP4000 User's Operation Manual.

VAC Volts Alternating Current

VARIABLE A 32-bit value that can contain a number used in a cue line. A variable may be assigned an ALIAS. There may be a maximum of 512 variables per system.

VDC Volts Direct Current

VERB Reserved words for the actions used in a CUE EVENT. The list of VERBs is in section 3 of the MP4000 User's Operation Manual.

VMR Vertical Module Rack.

.wav Extension used for WAVE files, contains sound data.

Z-RAILS Two Z-shaped rails on the back of the rack, adjacent and parallel to the space for the I/O connectors on the Modules when the modules are inserted. The field connectors attach to the Z-rails.

Index