Spectrum™

Release 8.3

Protocol Reference Guide

Rev J

Manual Part No. 28-0105

July 2017

Disclaimer

Harmonic reserves the right to alter the product specifications and descriptions in this publication without prior notice. No part of this publication shall be deemed to be part of any contract or warranty unless specifically incorporated by reference into such contract or warranty. The information contained herein is merely descriptive in nature, and does not constitute a binding offer for sale of the product described herein. Harmonic assumes no responsibility or liability arising from the use of the products described herein, except as expressly agreed to in writing by Harmonic. The use and purchase of this product does not convey a license under any patent rights, copyrights, trademark rights, or any intellectual property rights of Harmonic. Nothing hereunder constitutes a representation or warranty that using any product in the manner described herein will not infringe any patents of third parties.

Third-Party Product Trademarks

Adobe® After Effects®, Photoshop®, Flash® Professional, Premiere®

Avid® Media Composer®

Jünger Audio™

Apple® QuickTime®

Microsoft® Mediaroom®

Microsoft® PlayReady®

DOCSIS® 3.0

Start Over® TV

Dolby is a registered trademark of Dolby Laboratories. Dolby Digital, Dolby Digital Plus, Dolby Plus, aacPlus, AC-3, and Dolby® E are trademarks of Dolby Laboratories.

Level Magic and Jünger are trademarks of Jünger Audio Studiotechnik GmbH.

MPEG Audio technology licensed from Fraunhofer IIS http://www.iis.fraunhofer.de/amm/

PitchBlue® is a registered trademark of Vigor Systems.

QuickTime and the QuickTime logo are trademarks or registered trademarks of Apple Computer, Inc., used under license therefrom.

Third-Party Copyright Notes

Harmonic software uses version 3.15.4 of the FreeImage open source image library under FreeImage Public License (FIPL). See http://freeimage.sourceforge.net for details.

The product may include implementations of AAC and HE-AAC by Fraunhofer IIS; and MPEG Audio technology licensed from Fraunhofer IIS

The software described in this publication may use version 2.8 of FFmpeg open source package under Lesser General Public License (LGPL).

The software described in this publication is furnished under a nondisclosure agreement, or the License Agreement and Limited Warranty stated below, and the end user license agreement (which is furnished with the software), which may have additional terms. The software may be used or copied only in accordance with the terms of those agreements. By using the software, you acknowledge you have read the end user license agreement and the License Agreement and Limited Warranty provision.

The product described in this publication may be covered by one or more of U.S. Patents, their foreign counterparts and pending patent applications.

The product is distributed with certain other software that may require disclosure or distribution of licenses, copyright notices, conditions of use, disclaimers and/or other matter. Use of this product or otherwise fulfilling their conditions constitutes your acceptance of it, as necessary. Copies of such licenses, notices, conditions, disclaimers and/or other matter are available in any one of the following locations: the LEGAL NOTICES AND LICENSES section of the documentation directory of the product, user guide, or by contacting us at support@harmonicinc.com.

Notice

Information contained in this publication is subject to change without notice or obligation. While every effort has been made to ensure that the information is accurate as of the publication date, Harmonic Inc. assumes no liability for errors or omissions. In addition, Harmonic Inc. assumes no responsibility for damages resulting from the use of this guide.

License Agreement and Limited Warranty

1. AGREEMENT: This is a legal agreement ("Agreement") between you ("you" or "your") and Harmonic, or its appropriate local affiliate ("Harmonic", "we", "us" or "our"). Use of our product(s) and any updates thereto purchased or validly obtained by you (the "Products"), and/or the Software (as defined below) (collectively, the "System"), constitutes your acceptance of this Agreement. "Use" includes opening or breaking the seal on the packet containing this Agreement, installing or downloading the Software as defined below or using the Software preloaded or embedded in your System. As used herein, the term "Software" means the Harmonic owned software and/or firmware used in or with the Products and embedded into, provided with or loaded onto the Products in object code format, but does not include, and this Agreement does not address, any third-party or free or open source software separately licensed to you ("Third Party Software"). If you do not agree to this Agreement, you shall promptly return the System with a dated receipt to the seller for a full refund.

2. LICENSE: Subject to the terms and conditions of this Agreement (including payment), we hereby grant you a nonexclusive, nontransferable license to use the object code version of the Software embedded into, provided solely for use with or loaded onto the Product, and the accompanying documentation ("Documentation") for your internal business purposes. The Software and any authorized copies are owned by us or our suppliers, and are protected by law, including without limitation the copyright laws and treaties of the U.S.A. and other countries. Evaluation versions of the Software may be subject to a time-limited license key.

3. RESTRICTIONS: You (and your employees and contractors) shall not attempt to reverse engineer, disassemble, modify, translate, create derivative works of, rent, lease (including use on a timesharing, applications service provider, service bureau or similar basis), loan, distribute, sublicense or otherwise transfer the System, in whole or part except to the extent otherwise permitted by law. The Software may be operated on a network only if and as permitted by its Documentation. You may make one (1) back up copy of the object code of the Software for archival purposes only. Evaluation Software will be run in a lab, nonproductive environment. Results of any benchmark or other performance tests may not be disclosed to any third party without our prior written consent. Title to and ownership of the Software and Documentation, and all copyright, patent, trade secret, trademark, and other intellectual property rights in the System, shall remain our or our licensors' property. You shall not remove or alter any copyright or other proprietary rights notice on the System. We reserve all rights not expressly granted.

4. LIMITED WARRANTY: (a) Limited Warranty. We warrant to you that, commencing on your receipt of a Product and terminating 1 year thereafter, the System will perform substantially in accordance with its then-current appropriate Documentation. The Product (including replacements) may consist of new, used or previously-installed components. (b) Remedies. If the System fails to comply with such warranty during such period, as your sole remedy, you must return the same in compliance with our product return policy, and we shall, at our option, repair or replace the System, provide a workaround, or refund the fees you paid. Replacement Systems are warranted for the original System's remaining warranty period. (c) Exclusions. EVALUATION SOFTWWARE IS LICENSED ON AS-IS BASIS AND SUBJECT TO 4(d). We will have no obligation under this limited warranty due to: (i) negligence, misuse or abuse of the System, such as unusual physical or electrical stress, misuse or accidents; (ii) use of the System other than in accordance with the Documentation; (iii) modifications, alterations or repairs to the System made by a party other than us or our representative; (iv) the combination, operation or use of the System with equipment, devices, software or data not supplied by us; (v) any third party hardware or Third Party Software, whether or not provided by us; (vi) any failure other than by us to comply with handling, operating, environmental, storage or maintenance requirements for the System in the Documentation, including, without limitation, temperature or humidity ranges. (d) Disclaimers. We are not responsible for your software, firmware, information, or data contained in, stored on, or integrated with any Product returned to us for repair or replacement. SUCH LIMITED WARRANTY IS IN LIEU OF, AND WE SPECIFICALLY DISCLAIM, ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF SATISFACTORY QUALITY, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. WE DO NOT WARRANT THAT THE SYSTEM WILL MEET YOUR REQUIREMENTS OR BE UNINTERRUPTED OR ERROR-FREE. NO ADVICE OR INFORMATION, WHETHER ORAL OR WRITTEN, OBTAINED FROM US OR ELSEWHERE, WILL CREATE ANY WARRANTY NOT EXPRESSLY STATED IN THIS AGREEMENT. Some jurisdictions do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so such exclusions may not apply to you. In that event, such implied warranties or limitations are limited to 60 days from the date you purchased the System or the shortest period permitted by applicable law, if longer. This warranty gives you specific legal rights and you may have other rights which vary from state to state or country to country.

5. LIMITATION OF LIABILITY: WE AND OUR AFFILIATES, SUPPLIERS, LICENSORS, OR SALES CHANNELS ("REPRESENTATIVES") SHALL NOT BE LIABLE TO YOU FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, OR EXEMPLARY DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO LOST REVENUES, PROFITS OR SAVINGS, OR THE COST OF SUBSTITUTE GOODS, HOWEVER CAUSED, UNDER CONTRACT, TORT, BREACH OF WARRANTY, NEGLIGENCE, OR OTHERWISE, EVEN IF WE WERE ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGES. NOTWITHSTANDING ANY OTHER PROVISIONS OF THIS AGREEMENT, WE AND OUR REPRESENTATIVES' TOTAL LIABILITY TO YOU ARISING FROM OR RELATING TO THIS AGREEMENT OR THE SYSTEM SHALL BE LIMITED TO THE TOTAL PAYMENTS TO US UNDER THIS AGREEMENT FOR THE SYSTEM. THE FOREGOING LIMITATIONS SHALL NOT APPLY TO DEATH OR PERSONAL INJURY TO PERSONS OR TANGIBLE PROPERTY IN ANY JURISDICTION WHERE APPLICABLE LAW PROHIBITS SUCH LIMITATION. YOU ARE SOLELY RESPONSIBLE FOR BACKING UP YOUR DATA AND FILES, AND HEREBY RELEASE US AND OUR REPRESENTATIVES FROM ANY LIABILITY OR DAMAGES DUE TO THE LOSS OF ANY SUCH DATA OR FILES. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO SUCH EXCLUSIONS MAY NOT APPLY TO YOU.

6. CONFIDENTIALITY: Information in the System and the associated media, as well as the structure, organization and code of the Software, are proprietary to us and contain valuable trade secrets developed or acquired at great expense to us or our suppliers. You shall not disclose to others or utilize any such information except as expressly provided herein, except for information (i) lawfully received by the user from a third party which is not subject to confidentiality obligations; (ii) generally available to the public without breach of this Agreement; (iii) lawfully known to the user prior to its receipt of the System; or (iv) required by law to be disclosed.

7. SUPPORT: Updates, upgrades, fixes, maintenance or support for the System (an "Upgrade") after the limited warranty period may be available at separate terms and fees from us. Any Upgrades shall be subject to this Agreement, except for additional or inconsistent terms we specify. Upgrades do not extend the limited warranty period.

8. TERM; TERMINATION: The term of this Agreement shall continue unless terminated in accordance with this Section. We may terminate this Agreement at any time upon default by you of the license provisions of this Agreement, or any other material default by you of this Agreement not cured with thirty (30) days after written notice thereof. You may terminate this Agreement any time by terminating use of the System. Except for the first sentence of Section 2 ("License") and for Section 4(a) ("Limited Warranty"), all provisions of this Agreement shall survive termination of this Agreement. Upon any such termination, you shall certify in writing such termination and non-use to us.

9. EXPORT CONTROL: You agree that the Products and Software will not be shipped, transferred, or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions, or regulations (the "Export Laws"). You will indemnify, defend and hold us harmless from any and all claims arising therefrom or relating thereto. In addition, if the Products or Software are identified as export controlled items under the Export Laws, you represent and warrant that you are not a citizen, or otherwise located within, an embargoed nation (including without limitation Iran, Iraq, Syria, Sudan, Libya, Cuba, North Korea, and Serbia) and that you are not otherwise prohibited under the Export Laws from receiving the Software. All rights to the Products and Software are granted on condition that such rights are forfeited if you fail to comply with the terms of this Agreement.

10. U.S. GOVERNMENT RIGHTS: The Software and the documentation which accompanies the Software are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government as end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Harmonic, 4300 North First Street, San Jose, CA 95134 U.S.A.

11. GENERAL: You shall not assign, delegate or sublicense your rights or obligations under this Agreement, by operation of law or otherwise, without our prior written consent, and any attempt without such consent shall be void. Subject to the preceding sentence, this Agreement binds and benefits permitted successors and assigns. This Agreement is governed by California law, without regard to its conflicts of law principles. The U.N. Convention on Contracts for the International Sale of Goods is disclaimed. If any claim arises out of this Agreement, the parties hereby submit to the exclusive jurisdiction and venue of the federal and state courts located in Santa Clara County, California. In addition to any other rights or remedies, we shall be entitled to injunctive and other equitable relief, without posting bond or other security, to prevent any material breach of this Agreement. We may change the terms, conditions and pricing relating to the future licensing of our Systems and other intellectual property rights, including this Agreement, from time to time. No waiver will be implied from conduct or failure to enforce rights nor effective unless in a writing signed on behalf of the party against whom the waiver is asserted. If any part of this Agreement is found unenforceable, the remaining parts will be enforced to the maximum extent permitted. There are no third-party beneficiaries to this Agreement. We are not bound by additional and/or conflicting provisions in any order, acceptance, or other correspondence unless we expressly agree in writing. This Agreement is the complete and exclusive statement of agreement between the parties as to its subject matter and supersedes all proposals or prior agreements, verbal or written, advertising, representations or communications concerning the System.

Every reasonable attempt has been made to comply with all licensing requirements for all components used in the system. Any oversight is unintentional and will be remedied if brought to the attention of Harmonic at support@harmonicinc.com.

Documentation Conventions

This guide may use some special symbols and fonts to call your attention to important information. The following symbols appear throughout this guide:

DANGER: The Danger symbol calls your attention to information that, if ignored, can cause physical harm to you.

CAUTION: The Caution symbol calls your attention to information that, if ignored, can adversely affect the performance of your Harmonic product, or that can make a procedure needlessly difficult.

LASER DANGER: The Laser symbol and the Danger alert call your attention to information about the lasers in this product that, if ignored, can cause physical harm to you.

NOTE: The Note symbol calls your attention to additional information that you will benefit from heeding. It may be used to call attention to an especially important piece of information you need, or it may provide additional information that applies in only some carefully delineated circumstances.

IMPORTANT: The Important symbol calls your attention to information that should stand out when you are reading product details and procedural information.

TIP: The Tip symbol calls your attention to parenthetical information that is not necessary for performing a given procedure, but which, if followed, might make the procedure or its subsequent steps easier, smoother, or more efficient.

In addition to these symbols, this guide may use the following text conventions:

Convention	Explanation
`Typed Command`	Indicates the text that you type in at the keyboard prompt.
<Ctrl>, <Ctrl>+<Shift>	A key or key sequence to press.
Links	The *italics in blue* text to indicate Cross-references, and hyperlinked cross-references in online documents.
Bold	Indicates a button to click, or a menu item to select.
`ScreenOutput`	The text that is displayed on a computer screen.
Emphasis	The *italics* text used for emphasis and document references.

NOTE: You require Adobe Reader or Adobe Acrobat version 6.0 or later to open the PDF files. You can download Adobe Reader free of charge from www.adobe.com.

Contents

Introduction

The Spectrum™ System is a scalable storage architecture that combines a SAN (Storage Area Network) and NAS (Network Attached Storage) and that is designed to take on all creative challenges for today's digital media applications.

This document provides reference material relating to the Harmonic Spectrum System and its components as follows:

- *Introduction* provides an overview of the Spectrum System documentation suite, lists terms and conventions, and provides Technical Support information.
- *Command Sets* provides information regarding a variety of command sets and preroll parameters for controlling all MediaDirectors.
- *FTP Server Implementation on Spectrum MediaDecks and MediaDirector 2100, 2101 & 2012B* provides information regarding File Transfer Protocol (FTP) server implementation on Omneon MediaDeck™ and MediaDirector 2100, 2101, and 2102B.
- *FTP Server Implementation on Newer Spectrum Systems* provides information regarding File Transfer Protocol (FTP) server implementation on MediaDirector 2201 and 2202.

Spectrum System Documentation Suite

The table below describes the documents which comprise the Spectrum System Documentation Suite.

All items are packaged in self-extracting files and available for download from the Harmonic

This document...	Provides this information...
Spectrum System Installation Guide	■ System installation ■ Software installation and upgrade details ■ Orientation to system components ■ Troubleshooting system components ■ Specifications for system components
Spectrum Component Replacement Guides	Component replacement instructions for Spectrum devices
Spectrum Quick Reference Guides and the *Spectrum MediaDeck 7000 Installation Guide*	■ Front and back panel views of Spectrum devices ■ LED assignments and legends ■ Quick start steps
Spectrum X and ChannelPort Tools User Guide	■ Using FXTool and PreviewTool
Polaris Play: Playlist User Guide	■ Polaris Play: Playlist Control Overview ■ Using Polaris Play: Scheduler ■ Using Polaris Play: Playlist
Spectrum X and ChannelPort Template Authoring Guide	■ Spectrum X and ChannelPort template authoring
Spectrum Release Notes	Last minute information regarding a product release
Spectrum Media and Wrapper Formats	Supported clip types and wrapper formats
Spectrum MediaDeck 7000 Read Me First	■ Passwords for downloading MediaDeck and SystemManager files ■ Instructions for obtaining and installing the license file for SystemManager ■ Installation overview
Spectrum System Protocol Reference Guide	■ Command sets and preroll parameters for controlling Spectrum servers ■ The Harmonic implementation of FTP server

Software updates are available from the Harmonic website. Contact Harmonic Technical Support for login information.

The full download consists of the following:

- **Spectrum-v8.2.0.0-Software.zip**. This includes system software.
- **Spectrum-v8.2.0.0-Documentation.exe**. This includes product documentation.
- **HarmonicTemplatesAndTools-v8.2.0.0-SWandDoc.exe**. This includes the template authoring package, tools, and documentation, as well Polaris Play: Playlist Control tools and documentation.

Acrobat ® Reader® is needed to view the product documentation. Download this for free from: **http://www.adobe.com**

Technical Support

For information on contacting Harmonic Technical Support, refer to *Appendix A, Contacting the Technical Assistance Center*.

Useful Information when Contacting Technical Support

In order to assist Harmonic Technical Support, review the following information:

What version of firmware is installed on your system?

From the **Home** tab, click the **Upgrade Firmware** icon in the left-hand column to display the **Upgrade Firmware** page. The firmware version for each device is shown in the **Current Firmware Version** column.

What version of SystemManager software is installed?

From SystemManager, click the **Help** tab. The version is shown in the **Server Software** section of the page.

Which Windows operating system is running on the SystemManager client PC?

a. From Windows, click the **Start** button, and then click **Run**.

b. In the **Open** field, type: winver, and then press **Enter** to open the **About Windows** dialog box, which shows the version number.

How much memory is installed on the SystemManager platform? (for example, 256 MB, 512 MB, or 1 GB)

a. From Windows, click the **Start** button, and then click **Run**.

b. In the **Open** field, type: winver and then press **Enter** to open the **About Windows** dialog box. Look for the line which reads "Physical memory available to Windows."

Please provide the manager.oda file from the SystemManager platform or client PC

Harmonic Technical Support may request that you email the manager.oda file, which contains configuration information for your system. This file is located on the SystemManager platform at D:\Omneon\Manager\omdb, or if you are using a client PC with a single C: partition, it will be in the same directory on the C: drive.

What is the model and serial number of the hardware involved?

- For Spectrum and MediaDeck devices: from the **Home** tab, click the **Upgrade Firmware** icon in the left-hand column to display the **Upgrade Firmware** page. Both MediaDirectors and MediaDecks are listed in the **MediaDirectors** section. Find the Model Numbers and Serial Numbers listed in their respective columns.

  Scroll down to the **MediaPorts** section to view the Model Numbers and Serial Numbers for MediaPorts and MediaDeck Modules.
- For Harmonic MediaGrid Devices: Click the **Servers & Switches** icon in the left-hand column. From the Servers and Switches page, in the **Name** column, click the link for the Harmonic MediaGrid device to open the **Properties** page for that device.
- For ProXchange devices: Click the ProXchange Servers icon in the left-hand column. From the **Servers** page, in the **Name** column, click the link for the ProXchange device to open the **Properties** page for that device.
- For ProBrowse devices: Click the ProBrowse Servers icon in the left-hand column. From the **Servers** page, in the **Name** column, click the link for the ProBrowse device to open the **Properties** page for that device.

- ❑ For MAS devices: Click the MAS Servers icon in the left-hand column. From the Servers page, in the **Name** column, click the link for the MAS device to open the **Properties** page for that device.

For Spectrum Systems

What is the name of the Player that is being used?

From SystemManager, click the **Player Configuration** link in the left-hand column, and then click the name of the MediaDirector or MediaDeck. The **Player List** page for that device appears. The names and status of all players are listed.

What file format and bit rate is the Player configured for? (for example, MPEG, DV, IMX?)

a. From SystemManager, click the **Player Configuration** link in the left-hand column, and then click the name of the MediaDirector or MediaDeck. The **Player List** page for that device appears.

b. From the player list, click the **Properties** link to view all the details for a player.

If the problem is related to Ingest or Playout of a clip, what is the Clip ID involved?

The clip name or clip ID should be indicated by whatever software application you are using to play or record video. For ClipTool, clip names are displayed in the clip management area of the ClipTool main window.

- **What brand of Automation, if any, is being used for control?**
- **Is the Automation using VDCP or API for communication control?**
- **What other third party device (for example, Tandberg* or Snell and Wilcox*) is involved?**
- Please supply log files.

Chapter 1
Command Sets

This section provides information regarding a variety of command sets and preroll parameters for controlling all Spectrum MediaDirectors. Choose from the following topics:

VDCP

Choose from the following topics:

CmdSystem (0x00) and CmdVarIdSystem (0x08)

Command	Code
CmdDeleteProtectId	(0x15)
CmdUndeleteProtectId	(0x16)

CmdImmediate (0x01) and CmdVarIdImmediate (0x09)

Command	Code	Note
CmdStop	(0x00)	
CmdPlay	(0x01)	When playing clip1, with clip2 immediately following, the minimum preroll time needed to ensure that all fields within clip2 are decoded is 2 sec.
CmdRecord	(0x02)	

CmdStill	(0x04)	A “still” command on a port that is currently stilled (“still” port status bit set) and that has an ID cued (“cue/init-done” port status bit set) causes the port to advance to the cued ID and display the first frame. The “still” port status bit is cleared and the “cue/init-done” port status bit remains set.
CmdStep	(0x05)	
CmdContinue	(0x06)	
CmdJog	(0x07)	
CmdVariablePlay	(0x08)	

CmdPresetSelect (0x02) and CmdVarIdPresetSelect (0x0a)

Command	Code	Note
CmdRenameId	(0x1d)	
CmdClosePort	(0x21)	
CmdSelectPort	(0x22)	
CmdRecordInit	(0x23)	
CmdPlayCue	(0x24)	PlayCue loads the clip using the clips defaultIn and defaultOut points.
CmdCueWithData	(0x25)	
CmdDeleteId	(0x26)	
CmdPctToSignalFull	(0x2b)	
CmdRecordInitWithData	(0x2c)	
CmdDiskPrerollTime	(0x43)	When playing clip1, with clip2 immediately following, the minimum preroll time needed to ensure that all fields within clip2 are decoded is 2 sec.

CmdSenseRequest (0x03) and CmdVarIdSenseRequest (0x0b)

Command	Code	Notes
CmdOpenPort	(0x01)	The security mode parameter is ignored and the port is always opened UNLOCKED. Opening an opened port immediately streals the port from the original opener.
CmdNextIds	(0x02)	May exceed the 6 msec turn-around required by the protocol. Ids are reported in order of creation date.

CmdPortStatusRequest	(0x05)	
CmdPositionRequest	(0x06)	Position Request uses the player settings to select the reporting mode among drop525, nondrop525 and 625.
CmdActiveIdRequest	(0x07)	
CmdDeviceTypeRequest	(0x08)	
CmdSystemStatusRequest	(0x10)	The calculation of storage time uses the current Player's record bitrate. If a Player is not selected, the calculation uses the record bitrate of the last enabled VDCP input port. If there are no VDCP input ports, then 50 Mbps is used. The frames fields will always be zero. Drop/Non-drop setting has no effect.
CmdIdList	(0x11)	May exceed the 6 msec turn-around required by the protocol. Ids are reported in order of creation date.
CmdIdSizeRequest	(0x14)	Id Size Request uses the clip frame rate to select the reporting mode between drop525 and 625. If a player is selected and the player mode is nondrop525 and the clip is 29.97 Hz, then nondrop525 is selected.
CmdIdRequest	(0x16)	
CmdIdsAddedList	(0x18)	
CmdIdsDeletedList	(0x19)	

VDCP Extensions

The following information describes additional commands that have been implemented by Harmonic as a extension to the VDCP protocol. The additional commands are:

- *3X.70 / BX.70V_ID_INFO*
- *2X.60 / AX.60OPEN_USER_INFO*
- *2X.61CLOSE_USER_INFO*
- *2X.62ADD_USER_INFO*
- *3X.63GET USER INFO*

3X.70 / BX.70V_ID_INFO

The V_ ID_INFO command will return the Start of Material and Duration information for the specified file ID.

Command from Controller:

02	Byte Count	30	70	BITMAP	ID NAME (8 bytes)	Check Sum

or

02	Byte Count	B0	70	BITMAP	ID LENGTH	ID NAME	Check Sum

Where:

- ID LENGTH is a byte containing the size of the ID name (only used in the BX.70 command format).
- ID NAME is the ASCII character of the ID (8 characters for the 3X.70 format and up to 32 characters for the variable length BX.70 format).
- BITMAP is a byte containing a bitmap that specifies which information items will be returned. Only the ID Info 3 (Timecode) bit is used.

:

ID Info 8 (Not Used)	ID Info 7 (Not Used)	ID Info 6 (Not Used)	ID Info 5 (Not Used)	ID Info 4 (Not Used)	ID Info 3 (Timecode)	ID Info 2 (Not Used)	ID Info 1 (Not Used)

:

02	Byte Count	CMD ECHO	F0	BITMAP	BYTE 1	BYTE 2	-----	BYTE 8	Check Sum

02	Byte Count	CMD ECHO	F0	BITMAP	Check Sum

:Where:

- CMD ECHO matches the CMD1 byte of the command and will be either 30 or B0.
- BITMAP is a byte containing a bitmap that specifies which information items will be returned. Only the ID Info 3 (Timecode) bit is used.
- BYTE 1 through to BYTE 8 return the following values:

ID Information 3 – Timecode Information

Byte	MSB	LSB	Notes
Byte 1	FRAMES x 10	FRAMES x 1	Start of Material Timecode
Byte 2	SECONDS x 10	SECONDS x 1	Start of Material Timecode

Byte 3	MINUTES x 10	MINUTES x 1	Start of Material Timecode
Byte 4	HOURS x 10	HOURS x 1	Start of Material Timecode
Byte 5	FRAMES x 10	FRAMES x 1	Duration Timecode
Byte 6	SECONDS x 10	SECONDS x 1	Duration Timecode
Byte 7	MINUTES x 10	MINUTES x 1	Duration Timecode
Byte 8	HOURS x 10	HOURS x 1	Duration Timecode

Timecode values are returned in BCD format. No flags bits are set (such as the Drop Frame flag).

If the ID info 3 bit is set in the BITMAP and if the specified file cannot be found, then:

- The format of the reply will not change.
- The BITMAP will still be the same as the BITMAP byte in the command.
- The values of BYTE 1 through BYTE 8 will be 0.
- The *IdNotFound* bit is set in the port error status.

VDCP User Data Extensions

User Data will be stored in VALUE blocks within the file. Each VALUE block will be addressed by a KEYWORD name.

The KEYWORD must be an ASCII string; it has a variable length of up to 32 bytes. It does not need to have a terminating 0 since the command format includes a length value. A value of 0x00 is not allowed within the KEYWORD; all other values are allowed.

The VALUE block can hold arbitrary binary data; it has a variable length of up to 128 bytes.

User Data Commands

The following commands are available:

- *2X.60 / AX.60OPEN_USER_INFO*
- *2X.61CLOSE_USER_INFO*
- *2X.62ADD_USER_INFO*
- *3X.63GET USER INFO*

2X.60 / AX.60OPEN_USER_INFO

The OPEN_USER_INFO command will open the specified file ID for reading or writing User Data. If any file ID is already open for User Data by the communications port it will be closed.

Command from Controller:

02	BC	20	60	MODE	ID NAME (8 bytes)	CS

or

02	BC	A0	60	MODE	ID Length	ID NAME	CS

Where:

- MODE is 1 for read access, 2 for write access, 3 for read/write access.
- ID LENGTH is the size of the ID name (only used for the AX.60 command format).
- ID NAME is the ASCII character of the ID; it is either a fixed length of 8 bytes or a variable length.

If the specified file ID cannot be found, the *CueOrOperationFailed* bit is set in the port error status.

Return from Controlled Device:

ACK

2X.61CLOSE_USER_INFO

The CLOSE_USER_INFO command will close the file ID currently open for User Data.

Command from Controller:

02	02	20	61	CS

Return from Controlled Device:

ACK

2X.62ADD_USER_INFO

The ADD_USER_INFO command will add or replace the VALUE of the specified KEYWORD in the currently open file.

Command from Controller:

02	BC	20	62	KEYWORD LENGTH	KEYWORD	VALUE LENGTH	VALUE	CS

Where:

- KEYWORD LENGTH is a single byte containing the size of the KEYWORD.
- KEYWORD is a variable number of bytes containing the ASCII characters of the keyword (max. 32 characters). A value of 0x00 is not allowed within KEYWORD; any other value is allowed. KEYWORD does not need to have a terminator.
- VALUE LENGTH is a single byte containing the size of the VALUE.
- VALUE is a variable number of bytes containing the User Data (max. 128 bytes).

NOTE: An existing KEYWORD / VALUE set can be removed from the specified file by using this command that has the desired KEYWORD, has a VALUE LENGTH of 0, and has no VALUE bytes.

Return from Controlled Device:

ACK

3X.63GET USER INFO

The GET_USER_INFO command will return the VALUE from the specified KEYWORD in the currently open file. The keyword supplied with the command must be an exact match for an existing keyword that is stored within the specified file. The match is case sensitive. Another GET_USER_INFO command should not be started until after the first has completed.

Command from Controller:

02	BC	30	63	KEYWORD LENGTH	KEYWORD	CS

Where:

- KEYWORD LENGTH is a byte containing the size of the KEYWORD.
- KEYWORD is a variable number of bytes containing the ASCII characters of the keyword (max. 32 characters). A value of 0x00 is not allowed within KEYWORD; any other value is allowed. KEYWORD does not need to have a terminator.

Return from Controlled Device:

02	BC	30	E3	VALUE LENGTH	VALUE	CS

Where:

- VALUE LENGTH is a byte containing the size of the VALUE.
- VALUE is a variable number of bytes containing the User Data. (max. 128 bytes). If the keyword can't be found in the specified file, then:
- The VALUE LENGTH byte will be 0.
- There will be no VALUE bytes in the reply.
- The *SystemError* bit will be set in the port error status.

VDCP Enhancements For Back-to-Back Play of Short Clips

Standard VDCP makes it difficult for video servers to play back-to-back short clips. Normally, each successive ID to be played is cued with a cue command and then played with a play command. These cue and play commands alternate because:

- A cued ID does not play until a play command is received.
- Cueing another ID while an ID is already cued un-cues the previously cued ID and cues the new ID.
- At most one ID is playing while another ID is cued.

This procedure only allows for the server to know about the currently playing ID and the next ID to play. If the next ID to play has a short duration, the server does not get much time to process the ID after that.

Adding a new VDCP command enables a disk server to play a sequence of short clips. The command is scheduleID. This command essentially schedules an ID to play immediately following the current ID.

The scheduleID command requires one or three arguments: ID or ID, start position, and duration. Its response is ACK/NACK. The portStatus command must be used to determine if the scheduleID command completed successfully. The port will indicate cueInit in the port status while the scheduleID command is processing. An error while processing the scheduleID command will set one of the error bits in the port status.

This addition of this command is sufficient to play sequences of short clips. However, it adds to the complexity of the current state of a play port. Retrieving additional state information may be desirable and is achieved with a modification to the existing activeIdRequest command.

20.78/A0.78: scheduleID

The scheduleID command is used in conjunction with the standard playCue (or cueWithData) and the play commands. If the VDCP server port (the port) has an ID cued, then the scheduleID command schedules an ID to play immediately following the play of the cued ID. For example, let's assume ID xx1 is playing and ID xx2 is cued. A scheduleID command is issued for ID xx3. Play will pause at the end of ID xx1 if a play command is not issued before the end of ID xx1. A play command is now issued which causes ID xx2 to start playing. At this point, the port is in a play state and not cued. Issuing another play command would result in a cueNotDone error. ID xx2 plays to the end and ID xx3 automatically follows it. Play will pause at the end of ID xx3 if no further commands are issued.

The scheduleID command can also be used when the port is in a play and not cued state. In this state, a scheduleID command schedules the ID to play immediately following the currently playing ID. For example, let's assume ID xx1 is playing and no ID is cued. A scheduleID command is issued for ID xx2. This schedules ID xx2 to play immediately following ID xx1. Play will pause at the end of ID xx2 if no further commands are issued.

Multiple scheduleID commands can be issued in both the play and cued state. The successive commands simply add to the scheduled list of IDs to play. For example, IDs xx1 through xx5 could be played using the following command sequence: playCue xx1, play, scheduleID xx2, scheduleID xx3, scheduleID xx4, scheduleID xx5. Play would start after the play command and before the scheduleID xx2 command. The following command sequence would also play all five IDs: playCue xx1, scheduleID xx2, scheduleID xx3, play, scheduleIDxx4, scheduleID xx5. The difference is that the latter sequence tells the disk server more of what is to play before play starts.

NOTE: Mixing the usage of the scheduleID command while the port is cued or not cued is also allowed.

Another command sequence to play the above IDs would be: playCue xx1, scheduleID xx2, play, scheduleID xx3, playCue xx4, scheduleID xx5, play. The first play command will cause IDs xx1, xx2, and xx3 to play. The second play command would cause ID's xx4 and xx5 to play.

The playCue and cueWithData commands still work as usual in the sense that cueing an ID un-cues a currently cued ID. This un-cueing action also applies to any IDs that have been scheduled since the last playCue command. For example, let's assume the following command sequence has been issued: playCue xx1, play, playCue xx2, scheduleID xx3, and scheduleID xx4. The port is playing ID xx1, ID xx2 is cued, and IDs xx3 and xx4 will play immediately following the play of ID xx2. Now a playCue for ID xx5 is issued. This will un-cue ID xx2, unscheduled ID's xx3 and xx4, and cue ID xx5. A play command now will cause the ID xx5 to play.

A scheduleID command can only be issued while in the play or cued state. The cueInit bit in the port status is set upon receiving a valid scheduleID command. The cueInit bit is cleared upon completion of the scheduleID command. Additionally, a port status error bit may be set that would indicate a problem occurred. If an error bit is set, the ID is not scheduled. Processing a scheduleID command may set the port busy bit. Processing a scheduleID command does not block the processing of immediate commands.

Note the following usage:

- The cueInit bit is used to indicate ID cueing. It is set when processing a scheduleID, playCue or cueWithData command. The bit is cleared when all ID cueing is complete. These commands may be overlapped in which case the cueInit bit stays set until all cueing is complete. However, it is not recommended to overlap these commands because there are no error conditions to indicate which command failed. Coherent use of the error bits requires waiting for the cueInit bit to clear before issuing any of these commands.
- The cueInitDone bit is only set when a playCue or cueWithData command successfully cues an ID. It is generally cleared by the play command, however these commands also clear it: stop, step, continue, jog, variablePlay, and record.

Any one of several port status error bits may be set by the scheduleID command. In all cases the error is unrecoverable and the command is ignored. Following is a list of the error bits that may be set:

- portNotActive - Issuing the command while not playing or cued.
- wrongPortType - Issuing the command to a recording port.
- cmdWhileBusy - Issuing the command while the port is busy.
- illegalValue - Malformed ID, start or duration values.
- idNotFound – Failed to open ID.
- cueOrOperationFailed - An internal error.
- scheduleFull (extended ps3, 5th byte, 6th bit) – Too many IDs scheduled.

The timing requirements for a scheduleID command are disk server dependent. The disk server manufacturer should be able to specify something like the minimum time a scheduleID command should be issued before the ID is expected to play. For example, a Harmonic Server requires the scheduleID command roughly three seconds before the ID is to play.

Command (ID only format):

02	BC	20/A0	78	ID	CS

Command (ID plus start and duration format):

02	BC	20/A0	78	ID	SOM	DUR	CS

Where:

- ID is the eight-byte (20 case) or variable length (A0 case) ASCII encoded ID.
- SOM is the four-byte BCD encoded start position in frames, seconds, minutes, and hours.
- DUR is the four-byte BCD encoded duration in frames, seconds, minutes, and hours.

Response:

ACK

30.07 activeIdRequest

ActiveIdRequest is an existing command that takes no arguments. The behavior is modified with the addition of a single argument. Without the argument, it behaves as normal. The argument is a sequence number referring to an ID that is playing, scheduled, or cued. A sequence number of zero refers to the currently playing ID (or cued ID if not currently playing), a value of one refers to the next ID to play, a value of two to the ID after that and so on. A sequence number greater than or equal to the number of active IDs, returns a shortened response consisting only of the port state.

The response to an activeIdRequest command with the optional index argument contains the same information as a normal activeIdRequest response plus three more items. These additional items are: start position, duration, and state. The start position and duration items are the respective values specified when the ID was scheduled or cued. The state item describes the ID's scheduled versus cued state. The state item may take on the values described in the following table.

State Value	Definition
0	Currently playing (only one ID may have this value)
1	Scheduled to play (many IDs may have this value)
2	Cued (only one ID may have this value)
3	Scheduled after cue (many IDs may have this value)

Command:

02	03	30/BO	07	SN	CS

Where:

- SN is the sequence number

Response:

02	BC	30/BO	87	PS	ID	SOM	DUR	ST	CS

Where:

- PS is the one-byte port state, 0 for inactive, 1 for active
- ID is the eight-byte or variable length ID
- SOM is the four-byte BCD encoded start position in frames, seconds, minutes, and hours.
- DUR is the four-byte BCD encoded duration in frames, seconds, minutes, and hours.
- ST is the one-byte ID state

VDCP Enhancements for Improved Media Status Reporting

Starting with Release 5.4, enhancements are in place in the Spectrum software which improves the accuracy of media status reporting via VDCP on Spectrum and MediaDeck Servers. The enhancements improve the robustness of the video server by improving the accuracy of the information provided to automation clients when querying media state and cueing material for playout.

Prior to Release 5.4, Harmonic's implementation of VDCP only reported on the state of the clip metadata ("movie") file; and not if other parts of the clip – video, audio, and VBI essence – were missing. In Release 5.4 and later releases, the behavior and response to the VDCP ID Request command and the behavior of the VDCP Play Cue and Play Cue With Data commands has changed to support the reporting of these missing essence files.

Changes to ID Request (3X.16, BX.16)

Previously, the implementation of this command returned IN DISK if the clip metadata file existed, and NOT READY TO PLAY only if the clip was currently recording but had not recorded enough (based on information in the clip metadata) to start play-behind safely. Missing essence files did not alter the return value.

In Release 5.4 and later releases, the return value reports NOT READY TO PLAY if any of the essence files described in the clip metadata file are missing, or are of zero length. VDCP requires that the server responds to the client within 10ms; if the system is unable to determine, completely, the clip state in this time, QUERY PENDING is set and returned to the client. This causes the client to ask again for status to complete the request. The timing of these requests from the client prevents race conditions from occurring.

The system's first response to a new ID Request is QUERY PENDING. Further ID Request commands for the same ID will continue to respond QUERY PENDING for possibly several seconds. Once the states of all essence files for the ID are known, the response will clear QUERY PENDING and will indicate correct information (IN DISK, NOT READY TO PLAY, etc) as of that time.

If the server cannot immediately answer an ID Request then it will return QUERY PENDING and initiate an internal request for the information. Subsequent ID Requests for the same ID will eventually return a correct answer. An ID request for a different ID will also return QUERY PENDING but will not initiate an internal request. A subsequent request for this ID when an internal request is not already active generates an internal request. This serializes the internal work needed to satisfy ID Requests, in order to keep system load reasonable.

Automation systems must ensure that they follow up on a QUERY PENDING response in a timely manner, in order to prevent misleading or stale status. Thus, if automation gets a QUERY PENDING response it must ask again for the same ID within 4 seconds to get a non QUERY PENDING answer. If automation takes longer than 4 seconds to follow up, the internal process starts over for that ID. The server flushes the information once a non QUERY PENDING response is sent. This means another request for the same ID returns QUERY PENDING and the process starts over.

Changes to Play Cue (2X.24, AX.24) and Play Cue With Data (2X.25, AX.25)

Previously, the implementation of these commands set CUE DONE if the clip metadata file existed. (The state of CUE DONE was sensed using Port Status.) Missing essence files were undetected by the client and resulted in missing tracks playing black or silent. There was no mechanism available for the client to determine that tracks were missing from a playout.

In Release 5.4 and later releases, CUE DONE gets set only if all of the essence files described in the clip metadata file are present and have lengths greater than zero. Note that this does not guarantee that playout of all tracks will occur; the files could be corrupt, short, or otherwise unusable and this cannot be determined until play starts. As a result of this change, the time between the client's issuance of the Cue command, and achieving CUE DONE state may be longer than previously.

NOTE: If you have automation client(s) that require this functionality, be prepared to handle delays of several seconds or more between Cue and CUE DONE.

About Compatibility Switches

The changes described above can be controlled by two compatibility switches which are user-configurable. The default states for the switches should be the behavior prior to Release 5.4. One switch should govern the behavior of ID Request, and the other should control Play Cue and Play Cue With Data. Ensure that the switches can be set independently.

IMPORTANT: If you wish to avail of the VDCP enhancements described in this section, be prepared to set the switches as described above, and test and change automation system timing if indicated.

Sony Serial Protocol

Support of the Sony serial protocol facilitates the interconnection of MediaPorts to third party devices such as automation, archiving, and editing systems. Insert and assemble edit commands are not supported. Sony serial protocol functions include:

- **Timecode Query** — allows the timecode information to be registered by the editing application and lets the editing application issue a timecode query to get information about the current clip position (or timecode).
- **Play** — starts playing clips at 1.0x play speed (normal speed), in the forward direction.
- **Stop** — stops playback or recording at the current frame. The MediaPort switches to E-E.
- **Pause** — pauses a playing clip (even those in fast forward, record or rewind) and displays the current frame.
- **Fast Forward** — (full speed) moves instantly from the current mode to full fast forward (the clip shuttles forward at full speed and displays picture in shuttle).
- **Rewind** — (full speed) instantly changes from the current mode to full rewind (the clip rewinds at full speed and displays picture in shuttle).
- **Jog Forward** — (one frame) advances by one frame and then pauses.
- **Jog Reverse** — (one frame) backs up by one frame and then pauses.
- **Vari-Shuttle** — lets users incrementally change clip shuttle mode from full rewind to full fast forward speed.
- **Record** — starts recording into the current clip (which must have been created by the application through the API or through use of the Avid protocol extensions). All channels of video and audio will be recorded.
- **Eject** — removes (ejects) the clip from the Player and allows users to load a new clip from ClipTool, using the API, or using the Avid protocol extensions.

- **Mark In-point** — sets a user defined (marked) timecode value within the playback clip, usually the frame at which the user wants to begin digitizing. Clicking **Mark In** a second time (on the editing system) overwrites the old mark with a new one.
- **Mark Out-point** — sets a user defined (marked) timecode value within the playback clip, usually the frame at which the user wants to stop digitizing. Clicking **Mark Out** a second time (on the editing system) overwrites the old mark with a new one.
- **Go To Mark In** — cues the clip to the marked in-point. When the clip reaches the specified frame, it enters Pause mode.
- **Go To Mark Out** — cues the clip to the marked out-point. When the clip reaches the specified frame, it enters Pause mode.
- **Go To a Supplied Timecode** — cues the clip to a specified timecode. When the clip reaches the specified timecode, the clip enters Pause mode.
- **Data Sense and Current Time Sense commands** — for control and emulation purposes, provides the capability for retrieving all manner of status bits (not just timecode).
- **Avid Extended protocol** — allows (from within the editing application) the listing and loading of clips currently in the file system as well as creation and naming of new clips.

Setting Preroll Timing Parameters for Automation Systems Controlling Harmonic MediaDirector

To set preroll values in an automation system controlling a MediaDirector, refer to the automation system documentation. We recommend that you set the values for "disk preroll", "frames to send play early", and "frames to send record early" to 2 seconds.

Chapter 2
FTP Server Implementation on Spectrum MediaDecks and MediaDirector 2100, 2101 & 2012B

This section provides information regarding File Transfer Protocol (FTP) server implementation on Spectrum MediaDecks and MediaDirectors as follows:

- *Commands Definitions*
- *Transferring Clips Between Spectrum Systems*
- *Notes on Parsing QuickTime Files*
- *FTP Transfer While Recording*

NOTE: The FTP server is an independent implementation, created directly from the specification (RFC 959). The Harmonic FTP server follows the specification, RFC 959, which is an open standard available at: **http://www.ietf.org/rfc/rfc959.txt.** This chapter does not duplicate the specification, but notes instead where the Harmonic FTP server adheres to or diverges from the specification.

Commands Definitions

This section lists the FTP commands in an Spectrum System with MediaDecks and MediaDirector 2100, 2101, and 2102B in the following categories:

- *Access Control Commands*
- *Transfer Parameter Commands*
- *FTP Service Commands*
- *SITE Commands*

NOTE: For completeness, commands specified in RFC 959 but not implemented by Harmonic are identified as "not implemented."

Access Control Commands

Table 2–1 identifies the Access Control Commands used in Spectrum System with MediaDecks and MediaDirector 2100, 2101, and 2102B.

Table 2–1: Access Control Commands

Command(s)	Support
USER	Implemented as per RFC 959.
PASS	
CDW, CDUP, XCWD	
ACCT, SMNT, REIN	Not implemented.
QUIT	Implemented as per RFC 959.

USER

Implemented as per RFC 959. If domain level security in use, returns:

```
331, "User name "mrftp" in domain "studio", need password."
```

Otherwise returns:

```
230, "Welcome to Director. There are currently "3"users."
```

PASS

Implemented as per RFC 959. On success returns:

```
230, "User" mrftp" logged in, proceed..."
```

Returns these errors:

```
530, "User" mrftp"failed to log in, try again!"
```

CDW, CDUP, XCWD

Implemented as per RFC 959. On success returns:

```
250, "Current directory is now: "/fs/mydirectory" "
```

Returns these errors:

```
550, "Directory invalid"
553, "Absolute name too long"
```

ACCT, SMNT, REIN

Not implemented. Returns these errors:

```
500, "Command unrecognized"
```

QUIT

Implemented as per RFC 959. On success returns:

```
221, "Be seeing you"
```

Limitations for FTP Sessions

Table 2–2 shows the maximum number of FTP sessions according to MediaDirector and session type.

Table 2–2: FTP Session Limitations

Session Types	MCP 2101 MediaDirector	MCP 2102, 2102B MediaDirectors and MediaDecks
max. # store sessions	8	12
max. # retrieve sessions	10	12
max. # store/retrieve sessions	18	16
max. # login sessions	20	28

Transfer Parameter Commands

Table 2–3 identifies the Transfer Parameter Commands used in a Spectrum System with MediaDecks and MediaDirector 2100, 2101, and 2102B

Table 2–3: Transfer Parameter Commands

Command	Support
PORT	Implemented as per RFC 959.
PASV	Implemented as per RFC 959.
TYPE	Implemented as per RFC 959. Types A and I are supported.
STRU	Implemented as per RFC 959. File structure is supported.
MODE	Implemented as per RFC 959. **Stream Mode is supported.**

PORT

Implemented as per RFC 959. On success returns:

```
200, "PORT command successful."
```

Returns these errors:

```
501, "Failure on PORT command"
```

PASV

Implemented as per RFC 959. On success returns:

```
227, "Entering Passive Mode (10,35,74,26,4,20)".
```

Returns these errors:

```
425, "Cannot get listening connection",
425, "Cannot bind listening connection",
425, "Cannot listen connection"
```

TYPE

Implemented as per RFC 959. Types A and I are supported. On success returns:

```
200, "Using I for file transfers"
```

Returns these errors:

```
504, "Unimplemented parameter for TYPE, "C"",
```

STRU

Implemented as per RFC 959. File structure is supported. On success returns

```
200, "Using file structure".
```

Returns these errors:

```
504, "Unimplemented parameter for STRU, "F"",
```

MODE

Implemented as per RFC 959. Stream mode is supported. On success returns:

```
200, "Using stream mode"
```

Returns these errors:

```
504, "Unimplemented parameter for MODE, "T"",
```

FTP Service Commands

Table 2–4 identifies the FTP Service Commands used in a Spectrum System with MediaDecks and MediaDirector 2100, 2101, and 2102B. Commands used with systems comprised of MediaDirector 2201 and 2202 have been implemented according to RFC 959. Refer to: *http://www.ietf.org/rfc/rfc0959.txt?number=959*.

Table 2–4: FTP Service Commands

Command	Support
RETR <file>, STOR <file>	Implemented as per RFC 959.
STOU, APPE, ALLO	Not implemented.

Table 2–4: FTP Service Commands

REST	Implemented as per RFC 959.
RNFR <file>, RNTO <file>,	
ABOR	
DELE	
RMD <dir>, XRMD	
MKD <dir>, XMKD	
PWD, XPWD	
LIST, NLST	
SYST	
STAT	Not implemented.
HELP	Implemented as per RFC 959.
NOOP	
SIZE	
MDTM	

RETR <file>, STOR <file>

Implemented as per RFC 959. On success returns:

```
226, "File transfer completed."
```

Returns these errors:

```
553, "Absolute name too long"
550, "File "myfile"" is a directory",
550, "Could not open file "myfile"",
550, "Could not create file "myfile"",
550, "Only binary mode transfers allowed."
550, "Could not start data transfer task."
425, "Can't open data connection."
426, "Connection closed; transfer aborted."
452, "Requested action not taken. Insufficient storage space in
system."
451, "Requested action aborted: local error in processing."
226, "Abort successfully processed."
```

Additional information about STOR <File>:

Starting with Spectrum Release 4.7 SR4, support is available for an overwrite mode for STOR <file>. By default, STOR <file> does not overwrite files. To switch to overwrite mode, in a Telnet session enter:

```
>ftp overwrite [1:0]
```

Enter 1 to turn ON overwriting during file transmission. This is applied only to the specific MediaDirector host to which you connected. You will need to repeat the command for each additional host, if overwrite mode is required. The overwrite mode is N-V RAM and as such will survive a reboot.

STOU, APPE, ALLO

Not implemented. Returns this error:

```
500, "Command unrecognized."
```

REST

Implemented as per RFC 959. Returns this error:

```
501, "Failure on REST command"
```

Upon success returns:

```
350, "Restarting at ___ bytes" "Send RETR to initiate"
```

RNFR <file>, RNTO <file>

Implemented as per RFC 959. On success returns:

```
350, "Source file OK, waiting for RNTO"
250, "File rename successful"
```

Returns these errors:

```
503, "RNTO does not follow RNFR"
553, "Absolute name too long"
550, "Destination file "myfile" exists"
550, "File rename failed, "ox abcd", error”
550, "Source file "myfile" does not exist"
```

ABOR

Implemented as per RFC 959. On success returns:

```
226, "Abort successfully processed"
```

DELE

Implemented as per RFC 959. On success returns:

```
250, "File successfully deleted."
```

Returns these errors:

```
553, "Absolute name too long"
450, "Failure deleting file."
```

RMD <dir>, XRMD

Implemented as per RFC 959. On success returns:

```
257, "Directory "/fs/mydirectory" removed"
```

Returns these errors:

```
553, "Absolute name too long"
550, "Failure removing directory "/fs/mydirectory"",
```

MKD <dir>, XMKD

Implemented as per RFC 959. On success returns:

```
257, "Directory "/fs/mydirectory" created",
```

Returns these errors:

```
553, "Absolute name too long"
550, "Failure creating directory "/fs/mydirectory""
```

PWD, XPWD

Implemented as per RFC 959. On success returns:

```
257, "Current directory is: "/fs/mydirectory"",
```

LIST, NLST

Implemented as per RFC 959 on a separate connection. On success returns:

<directory listing>

```
226, "List succeeded."
```

Returns these errors:

```
553, "Absolute name too long"
550, "Name not found."
550, "Could not access name."
550, "Could not start data transfer task."
451, "Failure writing on socket."
451, "No memory."
425, "Can't open data connection."
426, "Connection closed; transfer aborted."
226, "Abort successfully processed."
```

SYST

Implemented as per RFC 959. On success returns:

```
213, "Omneon File System - stylelist format"
```

STAT

Not implemented: Returns these errors:

```
211, "Sorry, no status yet."
```

HELP

Implemented as per RFC 959. On success returns:

```
214, "Recognized commands:" <A list of commands>
214, "Commands marked with a * are not implemented."
```

Returns these errors:

```
501, "No specific help for command "Accio""
```

NOOP

Implemented as per RFC 959. On success returns:

```
200, "No operation completed okay."
```

SIZE

Implemented as per RFC 959. On success returns:

```
213 <file size>
```

Returns these errors:

```
553, "Absolute name too long"
550, "Could not open "/fs/mydirectory/myfile""
```

MDTM

Implemented as per RFC 3659. On success returns:

```
213 <file modify time>
```

Returns these errors:

```
553, "Absolute name too long"
550, "Could not open "/fs/mydirectory/myfile""
```

SITE Commands

Table 2–5 lists the Site Commands used in a Spectrum System

Table 2–5: Site Commands

Command	Support
HELP SITE	Implemented as per RFC 959.
SITE LISTFMT	
SITE STRIPE <type>	
SITE DELE <clip>	
SITE LIST <dir/clip>	
SITE RNFR <clip>, SITE RNTO<clip>	
SITE TIMEOUT <min>	
SITE RBW	

HELP SITE

Implemented as per RFC 959. On success returns:

```
214, "Recognized SITE commands:" <A list of commands>
214, ""
```

SITE LISTFMT

This command toggles the directory listing format returned in response to a LIST command. Each change affects the session making the change. Other sessions are not affected.

The default is UNIX style directory listing.

On success returns:

```
200, "Now using Unix list format"
200, "Now not using Unix list format"
```

File Striping and Associated Commands

The MediaDirector's file system supports three kinds of file stripe:

- Large stripe: Optimized for large bandwidth media types, for example video
- Small stripe: Optimized for smaller bandwidth media types, for example audio
- Blocked: Optimized for random access data

A file's stripe type is chosen when the file is created. FTP can be instructed by the SITE STRIPE command to select a particular stripe type when next creating a new file using the STOR command. Each change affects the session (login instance) making the change but other sessions are not affected. With no explicit STRIPE instruction, the FTP server selects the file stripe type based on the file name suffix.

SITE STRIPE <type>

The type parameter can have a value of LARGE, SMALL, or NO, or not be present. With no type parameter present, the current stripe type is returned.

On success returns:

```
200, "Current creation is NO_STRIPE"
200, "Current creation is now LARGE_STRIPE"
```

Returns these errors:

```
501, "Unknown argument to SITE STRIPE, "SUPER-SIZE""
```

Clip Manipulation and Associated Commands

The MediaDirector creates and uses video clips which consist of a QuickTime or MXF wrapper file and media files, including video, audio, and VBI. The MediaDirector will use video clips which are flattened into a single file and have internal media. Wrapper files are typically stored in the clip.dir and media files are typically stored in the sub directory clip.dir/media.dir. The names of the media files and their directories are held in the QuickTime .mov file. When a wrapper file is retrieved, from the Harmonic FTP server, for instance, the media file name and directory information is in the wrapper file and remains unchanged. When a wrapper file is stored to the Harmonic FTP server, as part of restoring a video clip from tape, for instance, the media files should be stored to the FTP server maintaining their names and directories. A problem occurs if there is a name conflict between the wrapper or one of the incoming media files and an existing media file on the server.

The following FTP SITE delete and rename commands allow clips to be transferred from an FTP client, for example an archive system, to the MediaDirector such that any media file name conflicts in the server's working directory will be resolved by the server. The process requires transferring the clips from the client to a temporary directory on the server, and then using the SITE rename-from and SITE rename-to command sequence to move the clip to the desired working directory. The Harmonic FTP server identifies any conflicts in media files, and renames the incoming files to avoid name conflicts with other existing media files. It modifies the wrapper file to accommodate the name changes, and it maintains the directory structure expressed in the wrapper file. In addition, a clip delete command is added.

The new SITE commands provide delete and rename functions on whole clips. They should only be used on a QuickTime, or MXF wrapper file (having the .mov suffix), and will read that file to determine which media files (eg. .mpg, .aiff) are entailed. These clip commands reuse the syntax of the existing ftp file commands for delete file and rename files.

SITE LIST <dir/clip>

The List command creates a file (.lst) listing all media files in the clip whose QuickTime wrapper file is <dir/clip>. The list file is created in the same directory <dir>. The name of the list file is the name of the QuickTime file <clip> with the suffix (.lst) added. If a list file already exists with that name, it is overwritten with the new list file. You can then use FTP to copy the list file to local storage to examine its contents, and delete the original on the Harmonic Spectrum System.

Following is a trace of the control connection of an FTP session that uses the SITE LIST command.

```
<220
 >USER mrftp
<230 Welcome. There are currently 1 users.
 >SYST
<213 Omneon File System - Unix style list format
 >CWD fs/clip.dir
<250 Current directory is now: "/fs/clip.dir"
 >PORT 10,1,1,108,18,245
<200 PORT command successful
 >LIST
<150 Opening data connection.
<226 List succeeded.
 >SITE LIST mpeg25.mov
<350 Listmedia OK, ready for retrieving
 >TYPE I
<200 Using binary for file transfers
 >PORT 10,1,1,108,18,249
<200 PORT command successful
 >RETR mpeg25.mov.lst
<150 Opening data connection.
<226 File transfer completed.
 >QUIT
<221 Be seeing you.
```

The contents of the list file can now be printed.

```
 > more mpeg25.mov.lst
/fs/clip.dir/mpeg25.mov
/fs/clip.dir/media.dir/mpeg25.m2v
/fs/clip.dir/media.dir/mpeg25.aiff
```

Returns these errors:

```
501 "Missing argument"
553 "Absolute name too long"
550 "Source clip does not exist"
450 "Failure finding source clip"
```

SITE DELE <clip>

The delete clip command will delete an entire clip, including the QuickTime wrapper file, and the media files pointed to by the wrapper file.

SITE RNFR <clip>, SITE RNTO<clip>

The rename-from clip command and the rename-to clip command are both required. These commands must appear in sequence, first the rename-from command, then the rename-to command (as with existing rename). The clip name can include a complete pathname, or it can state only the file name, leaving the full path left unstated, but implied to be the CWD path. All directories in both the rename-from and the rename-to path must exist.

Server Timeouts and Associated Commands

The Harmonic FTP server will conserve networking resources by timing out a connection if it fails to make progress, or stalls. Three cases exist. First a control connection goes silent, and no commands are sent to the FTP server. Second, a data transfer in either direction (store or retrieve) stalls because of the client, and the FTP server cannot complete the transfer. Third, a data connection expected by the FTP server (PASV) never arrives from the client. In all three cases, the default timeout is the same, and can be adjusted using the SITE timeout command.

SITE TIMEOUT <min>

The value of <min> is, first, the duration in minutes that the server will wait for a new command after a data transfer is completed, and second, the duration that the server will maintain a data connection while the transfer is stalled, and third, the duration the FTP server will hold a passive port open.

Valid values for the <min> parameter are from 1 minute up to 40 minutes. The default is 5 minutes. If the timeout command is sent by itself without any minute parameter the current timeout setting will be returned. The timeout value is local to the session setting it but other sessions are not affected. Within a session, every subsequent file transfer will use the new setting, either until the setting is changed, or until the session ends when the control connection terminates. Each new login to the Harmonic server starts with the default setting.

SITE RBW

This command gives the current setting for the FTP Transfer While Recording (Read Behind Write) function.

Returns

```
200, “Current read behind write setting: ON”
```

when the setting is on.

Returns

```
200, “Current read behind write setting: OFF”SITE
```

when the setting is off.

Enter:

```
SITE RBW 0
```

to turn the setting off.

Enter

```
SITE RBW 1
```

to turn the setting on (for that connection only).

NOTE: The default for FTP Transfer While Record is ON.

Transferring Clips Between Spectrum Systems

There are a variety of methods you can use to transfer clips from one MediaDirector to another. This section provides an example of using FTP to transfer a clip *directly between* MediaDirectors when *all* these conditions are met:

- The wrapper file format is QuickTime.
- The clip is an MPEG clip with a separate audio file(s).

Refer to *Transferring Clips Under Different Conditions* for suggestions on how to implement the transfer when conditions differ to those listed above.

For the purpose of this example, the following information is true:

- "fred.mov" is a QuickTime clip with media files “fred.mpg” and “fred.aiff”.
- The source MediaDirector has an IP address of: 10.35.74.26.
- The destination MediaDirector has an IP address of: 10.35.74.67.
- The source and destination directory for the clip is /FS1/clip.dir.
- **To delete a clip**: The API command OmPlrClipDelete, OmPlrOpen, and OmPlrClose are used.
- **To get a list of clips**: The API commands OmPlrClipGetFirst and OmPlrClipGetNext are used to enumerate a list of available clips. OmPlrOpen and OmPlrClose are also used.

NOTE: When transferring between MediaDirectors which are distant from each other, a VPN or other private network is necessary. The VPN is required so that the Harmonic API commands can be used; these commands utilize an RPC protocol. The VPN is also required so that the server ftp ports are not directly exposed to the Internet.

To transfer a Clip:

1. **Use the Harmonic Driver to obtain the necessary information as follows:**
   a. Query your clip library database to obtain the IP and directory for source device.
   b. Query your clip library database to obtain the IP and directory for destination device
   c. Obtain the clip name. Add a “.mov” extension if necessary. Combine with source directory.
2. **Obtain media names using omPlrLib.dll as follows:**
   a. Establish a connection to the source MediaDirector using OmPlrOpen and the source MediaDirector IP address.
   b. Confirm that the clip exists using OmPlrClipExists()
   c. Query for media names using OmPlrClipGetMediaNames(). Repeat the query as necessary.
   d. Close the connection using OmPlrClose.
3. **Setup a telnet session as follows:**
   a. To the *source* MediaDirector:

```
>telnet 10.35.74.26. 21
### port 21 is the FTP port
```

   b. To the *destination* MediaDirector:

```
>telnet 10.35.74.67. 21
### port 21 is the FTP port
```

4. **Setup a MediaDirector to MediaDirector FTP transfer for the first of the media files.** Refer to “RFC 959” and “RFC 765” for definitions of these commands.
   a. Send the following commands to the *source* MediaDirector:

```
>PASV
 ### expect a response of the form "227 Entering Passive Mode
10,35,74,26,4,20"
 ### which means it is waiting on port 4,20 (number varies)
for a response.
 ### Your app has to parse out the port number for use below.
>TYPE I
```

```
### response "200 Using binary for file transfers"
   >RETR /FS0/clip.dir/media.dir/fred.mpg
   ### retrieve off local disk; use full source path
```

b. Send the following commands to the *destination* MediaDirector:

```
   >PORT 10,35,74,26,4,20
### see response to PASV command for specific numbers to use
here
   >TYPE I
### response "200 Using binary for file transfers"
   >STOR /FS0/clip.dir/media.dir/fred.mpg
###  stores onto local disk; use full destination path & name of
choice
```

This is an alternate method with a passive destination:

a. Send the following commands to the destination MediaDirector:

```
>PASV
### expect a response of the form "227 Entering Passive Mode
10,35,74,67,4,20"
### Your app has to parse out the port number for use below.
>TYPE I
### response "200 Using binary for file transfers"
>STOR /FS0/clip.dir/media.dir/fred.mpg
### stores onto local disk; use full destination path & name of
choice
```

b. Send the following commands to the source MediaDirector:

```
>PORT 10,35,74,37,4,20
### see response to PASV command for specific numbers to use
here
>TYPE I
### response "200 Using binary for file transfers"
>RETR /FS0/clip.dir/media.dir/fred.mpg
   ### retrieve off local disk; use full source path
```

5. **Repeat the following steps for each of the remaining media files:**

a. Send the following commands to the *source* MediaDirector:

```
   >PASV
    ### expect a response of the form "227 Entering Passive Mode
10,35,74,26,4,20"
    ### Your app has to parse out the port number for use below.
   >RETR /FS0/clip.dir/media.dir/fred.aiff
```

b. Send the following commands to the *destination* MediaDirector:

```
   >PORT 10,35,74,26,4,20
    ### using port number parsed from PASV response of sending
MediaStore
   >STOR /FS0/clip.dir/media.dir/fred.aiff
```

This is an alternate method with a passive destination:

a. Send the following commands to the destination MediaDirector:

```
>PASV
### expect a response of the form "227 Entering Passive Mode
10,35,74,67,4,20"
### Your app has to parse out the port number for use below.
>TYPE I
```

```
### response "200 Using binary for file transfers"
>STOR /FS0/clip.dir/media.dir/fred.aiff
```

b. Send the following commands to the source MediaDirector:

```
>PORT 10,35,74,37,4,20
### see response to PASV command for specific numbers to use
here
>TYPE I
### response "200 Using binary for file transfers"
>RETR /FS0/clip.dir/media.dir/fred.aiff
```

6. **Initiate a** MediaDirector **to** MediaDirector **FTP transfer for the .mov file using these commands:**

a. Send the following commands to the source MediaDirector

```
   >PASV
    ### expect a response of the form "227 Entering Passive Mode
10,35,74,26,4,20"
    ### Your app has to parse out the port number for use below.
>TYPE I
### response "200 Using binary for file transfers"
   >RETR /FS0/clip.dir/fred.mov
```

b. Send the following commands to the destination MediaDirector:

```
   >PORT 10,35,74,26,4,20
    ### using port number parsed from PASV response of sending
MediaStore
>TYPE I
### response "200 Using binary for file transfers"
   >STOR /FS0/clip.dir/fred.mov
```

This is an alternate method with a passive destination:

a. Send the following commands to the destination MediaDirector:

```
>PASV
### expect a response of the form "227 Entering Passive Mode
10,35,74,67,4,20"
### Your app has to parse out the port number for use below.
>TYPE I
### response "200 Using binary for file transfers"
>STOR /FS0/clip.dir/fred.mov
```

b. Send the following commands to the source MediaDirector:

```
>PORT 10,35,74,37,4,20
### see response to PASV command for specific numbers to use
here
>TYPE I
### response "200 Using binary for file transfers"
>RETR /FS0/clip.dir/fred.mov
```

This completes the procedure.

Transferring Clips Under Different Conditions

Follow these suggestions to implement a clip transfer when conditions differ to the previous example.

- **If you wish to transfer a different media type(s)**: Follow the procedure above. Be aware that different file extension(s) to the example will be found and displayed when OmPlrClipGetMediaNames() is called.
- **If you wish to transfer a single clip with Internal Essence**: Follow the procedure above omitting steps 4 and 5 since only one file needs to be transferred.

Notes on Parsing QuickTime Files

The parsing of QuickTime files should only be attempted by those who need an independent code base that can handle the QuickTime file format. The following notes are intended to address QuickTime file format parsing in general, and specifically, the “alis” atom, which is used to hold pointers to external media files:

- You can download Apple's QuickTime player for free, and upgrade it to the pro version at a low cost. If you intend to create QuickTime files, you can use it to confirm that they were properly constructed. If you intend to read QuickTime files, you can use it to confirm resolution of externally referenced media files. An excellent method to test your code is to use the QuickTime player, create your own movie from a separate video and audio file, save the movie, and then try to parse the file. If you cannot parse it, then your code does not match Apple's.
- The Apple QuickTime file format specification is well written. Harmonic always strives to maintain compatibility with Apple's definition of QuickTime, so in general, if Apple's software can understand the QuickTime files produced by the QuickTime player or Final Cut Pro, then MediaDirectors should understand it as well. If this is not the case, contact Harmonic Technical Support for assistance. You can also use Apple's Dumpster to look at the low-level structure of a QuickTime file. If you are working on a PC or an Apple platform, consider using Apple's QuickTime API to read and write QuickTime files.
- Apple's open source code for the QuickTime streaming server parses QuickTime files, including the resolution of “alis” atoms. The streaming server source code is available at: **http://developer.apple.com/darwin/projects/streaming**.

  Go to QTAtom_dref::ResolveAlias() as a starting point.
- Since the correct tools are important, especially when trying to decipher binary files, Harmonic strongly suggest downloading a free copy of Bvi from: **http://bvi.sourceforge.net/**

  This allow you to more easily see where the atoms are located and see the encoding of the “pascal strings” used in QuickTime files. If you prefer “od”, you can get an up-to-date version that provides both a binary and text view from GNU here: **http://www.gnu.org/directory/GNU/textutils.html**.

FTP Transfer While Recording

This section details the FTP Transfer while Recording feature as follows:

- *Overview*
- *Features and Limitations*
- *Implementation*

Overview

The File Transfer Protocol (FTP) is a client/server protocol in which the client requests the server to transfer *whole* files. The FTP server will store and retrieve files at the request of an FTP client. Requests and server responses are carried through a control connection. Each file is transferred separately through a data connection; several files may be transferred independently and in parallel.

When retrieving a file, the FTP server reads the file and sends it across the network, in accordance with the FTP protocol. Internally, this read is implemented as a chunk-by-chunk read from the beginning to the end of the file. If the file changes subsequently, the transferred data represents an earlier version of the file. If the file changes while the FTP server is transferring it, because another entity is modifying or adding to it, a snapshot is transferred instead.

The "FTP Transfer While Recording" feature allows for the simultaneous creation/recording and retrieval of a file via FTP. Previously, neither the exact content, nor the length of the file could be predicted if a file was retrieved while it is being written to. With this feature, retrieving a file while it is being written to is supported under certain circumstances. Refer to *Features and Limitations* and *Implementation* for additional information. In particular, the file must be of the type "left to right".

This feature extends FTP to allow consistent retrieval of a file that is being created "left to right." If the file is being created more slowly than retrieval via FTP could run, then FTP will slow down and transfer data only as it becomes ready.

It is important to distinguish between a single file and all the files needed to play a video clip. Generally speaking, unless all the files of a clip are available, a video clip is not available. Thus, a video clip comprised of some files, which are not "left to right", could not make use of this feature.

NOTE: During storage and retrieval of a clip via FTP, each file is transferred using a separate FTP connection.

Features and Limitations

During a recording, a video clip is created in one of two forms:

- **Form 1**: A wrapper or meta file plus one or more media files, or,
- **Form 2**: A file containing both meta information and media (that is internal essence).

With Form 1 clips, there are several files to transfer. Typically the media files are large and the wrapper file is small.

With Form 2 clips, there is only one file to transfer. The files that make up a clip are either:

- **Type A**: created "left to right" or,
- **Type B**: created by "write then modify".

Type A files are only written/appended to at the end of the file; their size reflects stable data. Type B files do not follow those rules. If the file to transfer is Type A: "left to right," it can be retrieved via FTP while being created. However, if the file to transfer is Type B, it is not consistent until completely finished. These files cannot be retrieved via FTP while being created.

The QuickTime file format can be created in either a Form 1 wrapper file, or Form 2 wrapper file with included essence. However, since QuickTime files are typically Type B they cannot take advantage of this feature.

Video formats which support this feature include:

- D10
- MXF OP1a 25Hz with DV video and any kind of audio
- MXF OP1a 29.97Hz with DV video and no audio

Implementation

Keep the following important points in mind:

- This feature is always on.
- There can only be one writer to a file at a time. This is a requirement of the Harmonic file system. A file can be written by a client using FTP, or by a player as it is creating a clip through a record operation.
- With this feature, the FTP server retrieves a file, and when "End of File" (EOF) is encountered, checks if the file is open for write. If the file is open for write, the FTP server monitors the tail of the file and transfers data once it appears on disk. If the file is no longer open for write, the FTP server retrieves the rest of the file, and terminates retrieval once it encounters EOF.
- The FTP server monitors the tail of the file by sleeping and re-reading the file where it left off. As a stripe of data is written to disk, the FTP server will read it and transfer it. Network activity at this point will look like a flurry of packets following a wait of the sleep duration. Sleep time is set to 1 second. This value is adequate to notice file changes within one sleep period for most media bit rates, and 2 or 3 periods for VBI. FTP clients will typically wait forever for data to arrive. If the file is open for write, but no progress is made in the size of the file, the FTP server will timeout, stop monitoring the file, and conclude the retrieval. Timeout is set to 30 seconds. If the file is open for write, and progress is made but then stops, the FTP server will timeout, stop monitoring the file, and conclude the retrieval after 10 seconds. This value is required to avoid a false positive on media files (including VBI) during clip creation.

  Since Spectrum Release 4.7 SR4, support is available for configuring the FTP server timeout for files that stop after progress is made. Via a Telnet session enter:

  ```
  >ftp twrtimeout [no. of seconds]
  ```

- If the FTP server attempts to store a file from another FTP server while in passive mode, a timeout occurs if no data arrives from the other server. The default timeout is set to 5 minutes and log messages print after 20 seconds. Data must be received within 20 seconds to avoid engaging the Harmonic FTP server's timeout mechanism. This timing is required to support retrieval from another Harmonic FTP server, which is using the "FTP read while record" feature.
-

Chapter 3
FTP Server Implementation on Newer Spectrum Systems

This section provides information regarding File Transfer Protocol (FTP) server implementation on newer Spectrum MediaDirectors, ,which include MediaDirector 2201 and 2202, 2251 and 2252, MediaCenter, and MediaDeck 7000 as follows:

General Commands

Table 3–1 lists general FTP commands implemented for Spectrum MediaDirectors and the associated RFCs. Refer to *SITE Commands* for a list of Harmonic-specific SITE commands.

Command	Implemented per...
PASS	RFC 959
ALLO	
NOOP	
USER	
ACCT	
QUIT	
SYST	
TYPE	
MODE	
PASV	

Command	Implemented per...
STRU	RFC 959
HELP	
CWD, XCDW	
PORT	
PWD, XPWD	
CDUP, XCUP	
RETR	
REST	
STOR	
APPE	
STOU	
MKD, XMKD	
RMD, XRMD	
STAT	
NLIST, LIST	
ABOR	
MDTM	
SIZE	
FEAT	RFC 2389
OPTS	
EPRT	RFC 2428
ESTA	
ESTP	
EPSV	
SPSV	
PASV	RFC 2428
P@SW	
ALL	
MLST	RFC 3659
MLSD	

Table 3–1: General Commands

SITE Commands

Table 3–2 lists the Site Commands used in an Spectrum System.

Table 3–2: Site Commands

Command
SITE HELP
SITE STRIPE <type>
SITE DELE <clip>
SITE LIST <dir/clip>
SITE RNFR <clip>, SITE RNTO<clip>
SITE TIMEOUT <min>
SITE TWR [ON\OFF]
SITE TWRTO <sec>

SITE HELP

Implemented as per RFC 959. On success returns:

```
214, "Recognized SITE commands:" <A list of commands>
```

File Striping and Associated Commands

A file's stripe type is chosen when the file is created. FTP can be instructed by the SITE STRIPE command to select a particular stripe type when next creating a new file using the STOR command. Each change affects the session (login instance) making the change but other sessions are not affected. With no explicit STRIPE instruction, the FTP server selects the file stripe type based on the file name suffix.

SITE STRIPE <type>

The type parameter can have a value of LARGE, SMALL, NO, DEFAULT, or not be present. With no type parameter present, the current stripe type is returned. When DEFAULT is used, previous stripe types will be ignored and the normal way of choosing stripe type is applied.

On success returns:

```
200, "Current creation type is %s "
```

Returns these errors:

```
501, "Unknown argument to SITE STRIPE \"%s\"", str
```

Clip Manipulation and Associated Commands

The MediaDirector creates and uses video clips which consist of a QuickTime or MXF wrapper file and media files, including video, audio, and VBI. The MediaDirector will use video clips which are flattened into a single file and have internal media. Wrapper files are typically stored in the clip.dir and media files are typically stored in the sub directory clip.dir/media.dir. The names of the media files and their directories are held in the QuickTime .mov file. When a wrapper file is retrieved, from the Harmonic FTP server, for instance, the media file name and

directory information is in the wrapper file and remains unchanged. When a wrapper file is stored to the Harmonic FTP server, as part of restoring a video clip from tape, for instance, the media files should be stored to the FTP server maintaining their names and directories. A problem occurs if there is a name conflict between the wrapper or one of the incoming media files and an existing media file on the server.

The following FTP SITE delete and rename commands allow clips to be transferred from an FTP client, for example an archive system, to the MediaDirector such that any media file name conflicts in the server's working directory will be resolved by the server. The process requires transferring the clips from the client to a temporary directory on the server, and then using the SITE rename-from and SITE rename-to command sequence to move the clip to the desired working directory. The Harmonic FTP server identifies any conflicts in media files, and renames the incoming files to avoid name conflicts with other existing media files. It modifies the wrapper file to accommodate the name changes, and it maintains the directory structure expressed in the wrapper file. In addition, a clip delete command is added.

The new SITE commands provide delete and rename functions on whole clips. They should only be used on a QuickTime, or MXF wrapper file (having the .mov suffix), and will read that file to determine which media files (eg. .mpg, .aiff) are entailed. These clip commands reuse the syntax of the existing ftp file commands for delete file and rename files.

SITE LIST <dir/clip>

The List command creates a file (.lst) listing all media files in the clip whose QuickTime wrapper file is <dir/clip>. The list file is created in the same directory <dir>. The name of the list file is the name of the QuickTime file <clip> with the suffix (.lst) added. If a list file already exists with that name, it is overwritten with the new list file. You can then use FTP to copy the list file to local storage to examine its contents, and delete the original on the Harmonic Spectrum System.

Following is a trace of the control connection of an FTP session that uses the SITE LIST command.

```
<220
 >USER mrftp
<230 Welcome. There are currently 1 users.
 >SYST
<213 Omneon File System - Unix style list format
 >CWD fs/clip.dir
<250 Current directory is now: "/fs/clip.dir"
 >PORT 10,1,1,108,18,245
<200 PORT command successful
 >LIST
<150 Opening data connection.
<226 List succeeded.
 >SITE LIST mpeg25.mov
<350 Listmedia OK, ready for retrieving
 >TYPE I
<200 Using binary for file transfers
 >PORT 10,1,1,108,18,249
<200 PORT command successful
 >RETR mpeg25.mov.lst
<150 Opening data connection.
<226 File transfer completed.
 >QUIT
<221 Be seeing you.
```

The contents of the list file can now be printed.

```
 > more mpeg25.mov.lst
/fs/clip.dir/mpeg25.mov
/fs/clip.dir/media.dir/mpeg25.m2v
/fs/clip.dir/media.dir/mpeg25.aiff
```

Returns these errors:

```
501, "Missing argument to SITE LIST"
550, MSG_SANITY_FILE_FAILURE, name
550, MSG_SANITY_FILE_FAILURE, name/* name length */
```

SITE DELE <clip>

The delete clip command will delete an entire clip, including the QuickTime wrapper file, and the media files pointed to by the wrapper file.

Returns these errors:

```
501, "Missing argument to SITE DELE"
550, MSG_SANITY_FILE_FAILURE, name
550, MSG_SANITY_FILE_FAILURE, name/* name length */
```

SITE RNFR <clip>, SITE RNTO<clip>

The rename-from clip command and the rename-to clip command are both required. These commands must appear in sequence, first the rename-from command, then the rename-to command (as with existing rename). The clip name can include a complete pathname, or it can state only the file name, leaving the full path left unstated, but implied to be the CWD path. All directories in both the rename-from and the rename-to path must exist.

SITE RNFR

On success returns:

```
350, MSG_RENAME_RNFR_SUCCESS
```

Returns these errors:

```
0, MSG_RENAME_ABORT
550, MSG_ANON_CANT_RENAME
550, MSG_RENAME_FAILURE
550, MSG_SANITY_FILE_FAILURE, name
550, MSG_SANITY_FILE_FAILURE, name/* name length * /
550, MSG_FILE_DOESNT_EXIST
```

SITE RNTO

On success returns:

```
250, MSG_RENAME_SUCCESS
```

Returns these errors:

```
503, MSG_RENAME_NORNFR
550, MSG_ANON_CANT_RENAME
550, MSG_RENAME_FAILURE ": %S", sterror (errno)
550, MSG_SANITY_FILE_FAILURE, name
550, MSG_SANITY_FILE_FAILURE, name/* name length * /
```

Server Timeouts and Associated Commands

The Harmonic FTP server will conserve networking resources by timing out a connection if it fails to make progress, or stalls. Three cases exist. First a control connection goes silent, and no commands are sent to the FTP server. Second, a data transfer in either direction (store or retrieve) stalls because of the client, and the FTP server cannot complete the transfer. Third, a data connection expected by the FTP server (PASV) never arrives from the client. In all three cases, the default timeout is the same, and can be adjusted using the SITE timeout command.

SITE TIMEOUT <min>

The value of <min> is, first, the duration in minutes that the server will wait for a new command after a data transfer is completed, and second, the duration that the server will maintain a data connection while the transfer is stalled, and third, the duration the FTP server will hold a passive port open.

Valid values for the <min> parameter are from 1 minute up to 40 minutes for older Spectrum systems and 1 minute up to 60 minutes for newer Spectrum systems. The default is 5 minutes for older Spectrum systems and 15 minutes for newer Spectrum systems. If the timeout command is sent by itself without any minute parameter the current timeout setting will be returned. The timeout value is local to the session setting it but other sessions are not affected. Within a session, every subsequent file transfer will use the new setting, either until the setting is changed, or until the session ends when the control connection terminates. Each new login to the Harmonic server starts with the default setting.

On success returns:

```
200, “Current data control conn timeou is %d sec”
```

Returns this error:

```
501, “Argument to SITE TIMEOUT out of range \”%s\””
```

SITE TWR [ON\OFF]

This command gives or sets the current setting for the FTP Transfer While Recording function.

Returns

```
200, “Current transfer-while-record setting: %s”
```

when the setting is on.

Returns

```
200, “Current transfer-while-record setting: OFF”SITE
```

when the setting is off.

Enter:

```
SITE TWR OFF
```

to turn the setting off.

Enter

```
SITE RBW ON
```

to turn the setting on (for that connection only).

Returns this error:

```
501, “Argument to SITE TWR neither ON or OFF \”%s\””, onoff
```

NOTE: The default for FTP Transfer While Record is ON.

SITE TWRTO <sec>

Like TIMEOUT, this command gives the current value for timeout for the FTP Transfer While Recording function, or it allows you to set a different value. The default value is 10 seconds.

Returns:

```
200, “Current transfer-while-record timeout is %d secs”
```

Returns this error:

```
501, “Argument to SITE TWRTO out of range \”%s\””, sec
```

Transferring Clips Between Spectrum Systems

There are a variety of methods you can use to transfer clips from one MediaDirector to another. This section provides an example of using FTP to transfer a clip *directly between* MediaDirectors when *all* these conditions are met:

- The wrapper file format is QuickTime.
- The clip is an MPEG clip with a separate audio file(s).

Refer to *Transferring Clips Under Different Conditions* for suggestions on how to implement the transfer when conditions differ to those listed above.

For the purpose of this example, the following information is true:

- "fred.mov" is a QuickTime clip with media files “fred.mpg” and “fred.aiff”.
- The source MediaDirector has an IP address of: 10.35.74.26.
- The destination MediaDirector has an IP address of: 10.35.74.67.
- The source and destination directory for the clip is /FS1/clip.dir.
- **To delete a clip**: The Harmonic API command OmPlrClipDelete, OmPlrOpen, and OmPlrClose are used.
- **To get a list of clips**: The Harmonic API commands OmPlrClipGetFirst and OmPlrClipGetNext are used to enumerate a list of available clips. OmPlrOpen and OmPlrClose are also used.

NOTE: When transferring between MediaDirectors which are distant from each other, a VPN or other private network is necessary. The VPN is required so that the Harmonic API commands can be used; these commands utilize an RPC protocol. The VPN is also required so that the server ftp ports are not directly exposed to the Internet.

To transfer a Clip:

1. **Use the Harmonic Driver to obtain the necessary information as follows:**
   a. Query your clip library database to obtain the IP and directory for source device.
   b. Query your clip library database to obtain the IP and directory for destination device
   c. Obtain the clip name. Add a “.mov” extension if necessary. Combine with source directory.
2. **Obtain media names using omPlrLib.dll as follows:**
   a. Establish a connection to the source MediaDirector using OmPlrOpen and the source MediaDirector IP address.
   b. Confirm that the clip exists using OmPlrClipExists()
   c. Query for media names using OmPlrClipGetMediaNames(). Repeat the query as necessary.

d. Close the connection using OmPlrClose.

3. **Setup a telnet session as follows:**

a. To the *source* MediaDirector:

```
>telnet 10.35.74.26. 21
### port 21 is the FTP port
```

b. To the *destination* MediaDirector:

```
>telnet 10.35.74.67. 21
### port 21 is the FTP port
```

4. **Setup a MediaDirector to MediaDirector FTP transfer for the first of the media files.** Refer to "RFC 959" and "RFC 765" for definitions of these commands.

a. Send the following commands to the *source* MediaDirector:

```
>PASV
 ### expect a response of the form "227 Entering Passive Mode
10,35,74,26,4,20"
 ### which means it is waiting on port 4,20 (number varies)
for a response.
 ### Your app has to parse out the port number for use below.
>TYPE I
### response "200 Using binary for file transfers"
>RETR /FS0/clip.dir/media.dir/fred.mpg
### retrieve off local disk; use full source path
```

b. Send the following commands to the *destination* MediaDirector:

```
>PORT 10,35,74,26,4,20
### see response to PASV command for specific numbers to use
here
>TYPE I
### response "200 Using binary for file transfers"
>STOR /FS0/clip.dir/media.dir/fred.mpg
###  stores onto local disk; use full destination path & name of
choice
```

This is an alternate method with a passive destination:

a. Send the following commands to the destination MediaDirector:

```
>PASV
### expect a response of the form "227 Entering Passive Mode
10,35,74,67,4,20"
### Your app has to parse out the port number for use below.
>TYPE I
### response "200 Using binary for file transfers"
>STOR /FS0/clip.dir/media.dir/fred.mpg
### stores onto local disk; use full destination path & name of
choice
```

b. Send the following commands to the source MediaDirector:

```
>PORT 10,35,74,37,4,20
### see response to PASV command for specific numbers to use
here
>TYPE I
### response "200 Using binary for file transfers"
>RETR /FS0/clip.dir/media.dir/fred.mpg
### retrieve off local disk; use full source path
```

5. **Repeat the following steps for each of the remaining media files:**

a. Send the following commands to the *source* MediaDirector:

```
>PASV
 ### expect a response of the form "227 Entering Passive Mode
10,35,74,26,4,20"
 ### Your app has to parse out the port number for use below.
>RETR /FS0/clip.dir/media.dir/fred.aiff
```

b. Send the following commands to the *destination* MediaDirector:

```
>PORT 10,35,74,26,4,20
 ### using port number parsed from PASV response of sending
MediaStore
>STOR /FS0/clip.dir/media.dir/fred.aiff
```

This is an alternate method with a passive destination:

a. Send the following commands to the destination MediaDirector:

```
>PASV
### expect a response of the form "227 Entering Passive Mode
10,35,74,67,4,20"
### Your app has to parse out the port number for use below.
>TYPE I
### response "200 Using binary for file transfers"
>STOR /FS0/clip.dir/media.dir/fred.aiff
```

b. Send the following commands to the source MediaDirector:

```
>PORT 10,35,74,37,4,20
### see response to PASV command for specific numbers to use
here
>TYPE I
### response "200 Using binary for file transfers"
>RETR /FS0/clip.dir/media.dir/fred.aiff
```

6. **Initiate a** MediaDirector **to** MediaDirector **FTP transfer for the .mov file using these commands:**

a. Send the following commands to the source MediaDirector

```
>PASV
 ### expect a response of the form "227 Entering Passive Mode
10,35,74,26,4,20"
 ### Your app has to parse out the port number for use below.
>TYPE I
### response "200 Using binary for file transfers"
>RETR /FS0/clip.dir/fred.mov
```

b. Send the following commands to the destination MediaDirector:

```
>PORT 10,35,74,26,4,20
 ### using port number parsed from PASV response of sending
MediaStore
>TYPE I
### response "200 Using binary for file transfers"
>STOR /FS0/clip.dir/fred.mov
```

This is an alternate method with a passive destination:

a. Send the following commands to the destination MediaDirector:

```
>PASV
### expect a response of the form "227 Entering Passive Mode
10,35,74,67,4,20"
### Your app has to parse out the port number for use below.
```

```
>TYPE I
### response "200 Using binary for file transfers"
>STOR /FS0/clip.dir/fred.mov
```

b. Send the following commands to the source MediaDirector:

```
>PORT 10,35,74,37,4,20
### see response to PASV command for specific numbers to use
here
>TYPE I
### response "200 Using binary for file transfers"
>RETR /FS0/clip.dir/fred.mov
```

This completes the procedure.

Transferring Clips Under Different Conditions

Follow these suggestions to implement a clip transfer when conditions differ to the previous example.

- **If you wish to transfer a different media type(s)**: Follow the procedure above. Be aware that different file extension(s) to the example will be found and displayed when OmPlrClipGetMediaNames() is called.
- **If you wish to transfer a single clip with Internal Essence**: Follow the procedure above omitting steps 4 and 5 since only one file needs to be transferred.

Notes on Parsing QuickTime Files

The parsing of QuickTime files should only be attempted by those who need an independent code base that can handle the QuickTime file format. The following notes are intended to address QuickTime file format parsing in general, and specifically, the “alis” atom, which is used to hold pointers to external media files:

- You can download Apple's QuickTime player for free, and upgrade it to the pro version at a low cost. If you intend to create QuickTime files, you can use it to confirm that they were properly constructed. If you intend to read QuickTime files, you can use it to confirm resolution of externally referenced media files. An excellent method to test your code is to use the QuickTime player, create your own movie from a separate video and audio file, save the movie, and then try to parse the file. If you cannot parse it, then your code does not match Apple's.
- The Apple QuickTime file format specification is well written. Harmonic always strives to maintain compatibility with Apple's definition of QuickTime, so in general, if Apple's software can understand the QuickTime files produced by the QuickTime player or Final Cut Pro, then MediaDirectors should understand it as well. If this is not the case, contact Harmonic Technical Support for assistance. You can also use Apple's Dumpster to look at the low-level structure of a QuickTime file. If you are working on a PC or an Apple platform, consider using Apple's QuickTime API to read and write QuickTime files.
- Apple's open source code for the QuickTime streaming server parses QuickTime files, including the resolution of “alis” atoms. The streaming server source code is available at: **http://developer.apple.com/darwin/projects/streaming**

  Go to QTAtom_dref::ResolveAlias() as a starting point.
- Since the correct tools are important, especially when trying to decipher binary files, Harmonic strongly suggest downloading a free copy of Bvi from: *http://bvi.sourceforge.net/*

This allow you to more easily see where the atoms are located and see the encoding of the "pascal strings" used in QuickTime files. If you prefer "od", you can get an up-to-date version that provides both a binary and text view from GNU here: **http://www.gnu.org/directory/GNU/textutils.htm**/

FTP Transfer While Recording

This section details the FTP Transfer while Recording feature as follows:

- *Overview*
- *Features and Limitations*
- *Implementation*

Overview

The File Transfer Protocol (FTP) is a client/server protocol in which the client requests the server to transfer *whole* files. The FTP server will store and retrieve files at the request of an FTP client. Requests and server responses are carried through a control connection. Each file is transferred separately through a data connection; several files may be transferred independently and in parallel.

When retrieving a file, the FTP server reads the file and sends it across the network, in accordance with the FTP protocol. Internally, this read is implemented as a chunk-by-chunk read from the beginning to the end of the file. If the file changes subsequently, the transferred data represents an earlier version of the file. If the file changes while the FTP server is transferring it, because another entity is modifying or adding to it, a snapshot is transferred instead.

The "FTP Transfer While Recording" feature allows for the simultaneous creation/recording and retrieval of a file via FTP. Previously, neither the exact content, nor the length of the file could be predicted if a file was retrieved while it is being written to. With this feature, retrieving a file while it is being written to is supported under certain circumstances. Refer to *Features and Limitations* and *Implementation* for additional information. In particular, the file must be of the type "left to right".

This feature extends FTP to allow consistent retrieval of a file that is being created "left to right." If the file is being created more slowly than retrieval via FTP could run, then FTP will slow down and transfer data only as it becomes ready.

It is important to distinguish between a single file and all the files needed to play a video clip. Generally speaking, unless all the files of a clip are available, a video clip is not available. Thus, a video clip comprised of some files, which are not "left to right", could not make use of this feature.

NOTE: During storage and retrieval of a clip via FTP, each file is transferred using a separate FTP connection.

Features and Limitations

During a recording, a video clip is created in one of two forms:

- **Form 1**: A wrapper or meta file plus one or more media files, or,
- **Form 2**: A file containing both meta information and media (that is internal essence).

With Form 1 clips, there are several files to transfer. Typically the media files are large and the wrapper file is small.

With Form 2 clips, there is only one file to transfer. The files that make up a clip are either:

- **Type A**: created “left to right” or,
- **Type B**: created by “write then modify”.

Type A files are only written/appended to at the end of the file; their size reflects stable data. Type B files do not follow those rules. If the file to transfer is Type A: “left to right,” it can be retrieved via FTP while being created. However, if the file to transfer is Type B, it is not consistent until completely finished. These files cannot be retrieved via FTP while being created.

The QuickTime file format can be created in either a Form 1 wrapper file, or Form 2 wrapper file with included essence. However, since QuickTime files are typically Type B they cannot take advantage of this feature.

NOTE: The *only* format which supports this feature is MXF OP1a Low Latency.

Implementation

Keep the following important points in mind:

- This feature is always on.
- There can only be one writer to a file at a time. This is a requirement of the Harmonic file system. A file can be written by a client using FTP, or by a player as it is creating a clip through a record operation.
- With this feature, the FTP server retrieves a file, and when “End of File” (EOF) is encountered, checks if the file is open for write. If the file is open for write, the FTP server monitors the tail of the file and transfers data once it appears on disk. If the file is no longer open for write, the FTP server retrieves the rest of the file, and terminates retrieval once it encounters EOF.
- The FTP server monitors the tail of the file by sleeping and re-reading the file where it left off. As a stripe of data is written to disk, the FTP server will read it and transfer it. Network activity at this point will look like a flurry of packets following a wait of the sleep duration. Sleep time is set to 1 second. This value is adequate to notice file changes within one sleep period for most media bit rates, and 2 or 3 periods for VBI. FTP clients will typically wait forever for data to arrive. If the file is open for write, but no progress is made in the size of the file, the FTP server will timeout, stop monitoring the file, and conclude the retrieval. Timeout is set to 75 seconds. If the file is open for write, and progress is made but then stops, the FTP server will timeout, stop monitoring the file, and conclude the retrieval after 25 seconds. This value is required to avoid a false positive on media files (including VBI) during clip creation.

  Use the SITE TWRTO command to configure the FTP server timeout for files that stop after progress is made.
- If the FTP server attempts to store a file from another FTP server while in passive mode, a timeout occurs if no data arrives from the other server. The default timeout is set to 15 minutes.

Limitations for FTP Sessions

Table 3–3 shows the maximum number of FTP sessions supported.

Table 3–3: Maximum Number of FTP Sessions Supported

Session Types	MediaDirector 2201 and 2202
max. # store sessions	Total of 60 for all session types. (Note: There is a separate limit of 48 *Media API* Active Transfers per MediaDirector 2201/2. Active Transfers are transfers between the MediaDirector 2201/2 and the Harmonic MediaGrid.)
max. # retrieve sessions	
max. # store/retrieve sessions	
max. # login sessions	

About FTP Pattern Matching on Spectrum MediaDirectors

The manner in which FTP servers perfom pattern matching on pathnames varies according to MediaDirector type. Newer MediaDirectors handle pathnames with characters like [ ], { }, and ( ) in a different manner to older MediaDirectors such as MediaDirector 2100, 2101, 2102, and 2102B. *Table 3–4* provides a list of characters and how they are matched according to MediaDirector type.

Table 3–4: FTP Pattern Matching According to MediaDirector

Character/ Pattern Match	MediaDirector 2100, 2101, 2102, and 2102B	Newer MediaDirectors
?	Matches any single character	
*****	Matches any set of contiguous characters	
[range]	Matches a single character within the range, for example [a-z] or [0-9]	
****	Literal	This is the escape character which makes the next character the *actual* character, but only for certain characters
\[	Literal	[
]	Literal, as long as there is no preceeding [	
\\{	Literal	{
{	Literal, no special meaning	Ignored if in a pair of {}, otherwise literal
}	Literal, no special meaning	Literal, as long as there is no preceding {
(	Literal, no special meaning	
)		

Appendix A
Contacting the Technical Assistance Center

Harmonic Global Service and Support has many Technical Assistance Centers (TAC) located globally, but virtually co-located where our customers can obtain technical assistance or request on-site visits from the Regional Field Service Management team. The TAC operates a Follow-The-Sun support model to provide Global Technical Support anytime, anywhere, through a single case management and virtual telephone system. Depending on time of day, anywhere in the world, we will receive and address your calls or emails in one of our global support centers. The Follow-the-Sun model greatly benefits our customers by providing continuous problem resolution and escalation of issues around the clock.

Report an issue online at:

http://harmonicinc.com/webform/report-issue-online

Table A–1: Technical Support Phone Numbers and Email Addresses

Region	Telephone Technical Support	E-mail
Americas	888.673.4896 (888.MPEG.TWO) or +1.408.490.6477	*support@harmonicinc.com*
Europe, Middle East, and Africa	+44.1252.555.450	*emeasupport@harmonicinc.com*
India	+91.120.498.3199	*apacsupport@harmonicinc.com*
Russia	+7.495.926.4608	*rusupport@harmonicinc.com*
Mainland China	+86.10.6569.5580	*chinasupport@harmonicinc.com*
Japan	+81.3.5565.6737	*japansupport@harmonicinc.com*
Asia Pacific – Other Territories	+852.3184.0045 or 65.6542.0050	*apacsupport@harmonicinc.com*

The Harmonic Inc. support website is:

http://www.harmonicinc.com/content/technical-support

The Harmonic Inc. software download locations are:

All Harmonic software except Cable Edge software	Software updates are available from the Harmonic website. Contact Harmonic Technical Support for login information.
Cable Edge software	*ftp://ftp.harmonicinc.com*

The Harmonic Inc. corporate address is:

Harmonic Inc.
4300 North First St.
San Jose, CA 95134, U.S.A.
Attn: Customer Support

The corporate telephone numbers for Harmonic Inc. are:

Tel. 1.800.788.1330 (inside the U.S.)
Tel. +1.408.542.2500 (outside the U.S.)
Fax.+1.408.542.2511