Motion Engineering

33 South La Patera Lane
Santa Barbara, CA 93117-3214
ph (805) 681-3300
fax (805) 681-3311
tech@motioneng.com
www.motioneng.com

Release Note

MPI Library and Firmware

for XMP and ZMP Controllers

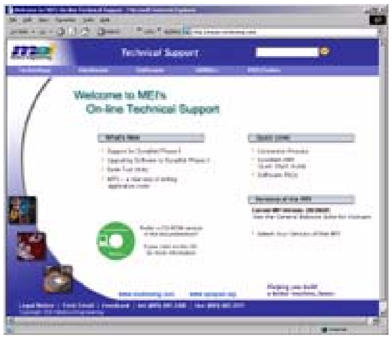
MPI Library Version: 20031222
Firmware Version: 514A6

Revised 22Dec2003
DCR 729

1 Introduction

Welcome to the latest release of Motion Engineering's Motion Programming Interface Library. This distribution has been prepared for Windows® NT/2000/XP. The distribution was built using Visual C++ v6.0 and tested using Visual C++ v6.0. This document provides an overview of the release, and describes the new features, changes, and bug fixes between the following versions:

	New Version	Previous Version
MPI Library	20031222	20031025
Firmware	514A6	510A3
Motion Console	03.40.05	03.40.04
Motion Scope	01.21.11	01.21.09



Technical Support

Welcome to MEI's On-line Technical Support

http://support.motioneng.com

MEI'S ON-LINE DOCUMENTATION SYSTEM

Featuring:

- Up-to-Date Information
- Keyword Search Engine
- SynqNet Documentation
- Dynamic Navigational Menu
- Print-friendly PDFs

Guaranteed to help you find
the **right** information...***faster!***

This software release completes the features to support pulse drives with SynqNet nodes. This release also contains preliminary support for the new ZMP-Series controllers. They are available for demonstration purposes only. If you are interested, please contact MEI.

Node FPGAs

This release includes new SynqNet Node FPGA Runtime images, Version 302. These contain some internal changes to the secondary encoder logic, support for Node ADC's, and enhancements to the pulse engine. Please see the FPGA Release Notes.pdf for more details. The MPI will return a WARNING message if a Node does not have a compatible Runtime image. Also, the MPI's SynqNet Info structure reports if the Runtime image is the "Default" image that was included with the software release. MEI always recommends that you use the FPGA images included with the software release. Please see issue MPI 1227 in this document for more information about the MPI's version check.

Pulse

Pulse support has been added to the RMB-10v2 FPGA, XMP controller firmware, MPI, and Motion Console. The pulse outputs support stepper drives and pulse controlled servos with either Step/Dir, CW/CCW, or QuadA/B logic. The maximum pulse rate is 5MHz. The pulse width is configurable from 100 nanoseconds to 1 millisecond. A future release will extend the maximum pulse rate to 10MHz.

When configuring the pulse width, make sure the maximum commanded velocity does not generate a pulse period that is less than 2x the pulse width. For example, if the maximum velocity is 100 kHz, the pulse width must not be greater than 5 microseconds. If the pulse width is too large for the commanded velocity, unstable behavior will result.

When using pulse outputs, the quadrature encoder inputs are still available. The position feedback information is not used to correct the pulse motor's position, but can be used by an application to monitor the actual position of the motor. If the encoder feedback is not in a one-to-one ratio with the step motor resolution, be sure to disable the position error limit action. If you do not have an encoder, the feedback can be simulated by enabling the motor's loopback feature.

The pulse feature also includes packet error correction. Normally, the controller sends a pulse command each sample period. If the node fails to receive the pulse command, it will continue to generate pulses at the last commanded velocity. The node will keep track of the pulses it generates without a corresponding command from the controller and will send the pulse count back to the controller. The controller uses the pulse count to correct the number of pulses over the next 4 controller periods. Using this technique, a bad command packet will not cause a significant discontinuity in the pulse control outputs to the drive. If there is a serious network communication problem, the packet error fault limit will trigger a network fault recovery. If the network communication problem is really serious, the packet error fail limit will trigger a node shutdown, causing the pulse generation to immediately halt.

ZMP

The ZMP-Series is a new high performance SynqNet controller. It supports the same features as the XMP-Series and uses the same MPI Library. The firmware file is compiled from the same sources as the XMP-Series, but the filename uses a "z" prefix. For example, the default ZMP-Series firmware would be zmp514a6.bin. Since the ZMP-Series controllers have more power and memory, they will eventually support more features than the XMP-Series. Both controller series will continue to use the same MPI library. If you are interested in evaluating a ZMP controller, please contact MEI.

If you are using a ZMP-Series controller (not XMP-Series), there are a few open issues with this release:

> **Reset** - Do not reset the controller while a second application is reading/writing from the SynqNet or SqNode objects. For example, if Motion Console is open while the reset.exe utility is run, the controller will lock-up. To recover, turn the power off/on to the host computer. This problem will be corrected in a future release.
>
> **Upgrading Firmware** - If you have a ZMP-Series controller and are using a previous software release (20031025 / 510A3), please contact MEI. There is a special one time procedure that is needed to upgrade the controller's firmware to this (and later) releases.

Device Driver

The MEIXMP.SYS device driver has been updated to support the new ZMP-Series controllers. See issue MPI 1242 in this document for more information.

Due to the changes, the MEIXMP.SYS device driver is not compatible with older software releases. If you need to downgrade to an older software release, you will need to manually replace the MEIXMP.SYS and MEIXMP.INF files in your WinNT\System32\drivers directory. Be sure to disable the device driver before replacing the .sys/.inf files, and then re-enable the device driver to load the new files.

1.1 System Requirements

1.1.1 Operating System

The MPI release is built to operate on Windows® NT 4.0 / 2000 / XP.

1.1.2 Visual C++ DLLs

The MPI is built using Microsoft Visual C++ 6.0.

1.2 Installing the Distribution

⚠ **WARNING! You must reboot your system!**
If you have not used a InstallShield for Windows Installer program before, the MEI Install Shield will need to install InstallShield installer files before actually installing the MDK. You will have to **reboot** your system after these files are installed. Please shut down all programs before running the InstallShield for the first time.

⚠ **WARNING!** If you are upgrading from a previous MPI/XMP software release, **you will need to remove or archive all previous releases**. This will prevent any conflicts between old and new files. To remove the previous MPI/XMP software release, select **Start -> Control Panel -> Add/Remove Programs**. Select the **MPI/XMP Development Toolkit** entry and click on the **Add/Remove** button.
NOTE: The MPI/XMP software release can also be removed by running the MDK InstallShield and choosing the remove option.

The MPI/XMP distribution comes in two parts. The first part is an InstallShield distribution. Key components of the distribution are:

- device driver (meixmp.sys for WinNT / Win2000 / WinXP)
- firmware
- MPI dynamic link library
- utilities
- sample applications

To install the MPI/XMP software release, insert the MDK CD-ROM. To start the set-up process, click on WinNTSetup.exe. Follow the InstallShield instructions. The InstallShield will take care of installing the DLL and will also set the PATH environment variable to XMP\bin\WinNT for WinNT, XMP\bin\Win2000 for Win2000, and XMP\bin\WinXP for XP under the default installation directory.

The second component of this distribution contains customer-specific applications and files. This is provided to you in a separate InstallShield. To install this custom component, click on the InstallShield and follow the instructions.

2 General Changes / New Features

This section lists the changes since the 20030620.1.1 production release, beginning with the most recent.

⚠ - Denotes modifications that may require changes in code.

Version 20031222

	New Version	Previous Version
Firmware	514A6	510A3
MPI Library	20031222	20031025

2.1 Motor Pulse Status Bits MPI 1268

The following has been added to meiMotorStatus in stdmei.h:

```
typedef struct MEIMotorStepperStatus {
     long    pulseLockLost;     /* TRUE if Pulse jitter logic has lost lock,
                                 otherwise FALSE */
     long    pulseStatus;       /* TRUE if a Pulse was generated incorrectly,
                                 otherwise FALSE */
} MEIMotorStepperStatus;
```

pulseLockLost - Indicates that the pulse module lost timing lock with the SynqNet network.

pulseStatus - Indicates that the pulse module generated a new pulse when the previous pulse was not finished. It usually indicates that the pulse width is incorrect for the desired velocity.

2.2 Support for Soonhan SynqNet-RMB MPI 1263

Support has been added for Soonhan's new SynqNet-RMB series:

Model #	PN	Description
IO2K-SQA2	TQ01-0000	Analog 2-Axis without Aux.DAC, ADC
IO2K-SQP2	TQ00-0000	Pulse 2-Axis
IO2K-SQP2C	TQ00-0001	Pulse 2-Axis without feedback, GPIO

2.3 Cable Length Check Changes MPI 1262

In previous versions of the MPI, the network cable length mismatch check was always enabled. If enabled, the cable length check is performed during network initialization. The cable length check is now enabled/disabled using meiSynqNetConfigSet/Get(...) methods. Suggested minimum and maximum values (based on discovered cable lengths), can be obtained using meiSynqNetInfo(...). Use these values, or your own values to enable the cable length check. To disable the cable length check, set all min, max, and nominal cable length values to zero.

The network topology must be saved to flash before saving cable lengths to flash memory. Use meiSynqNetFlashTopologySave(...) to save the topology to controller flash memory.
meiSynqNetFlashTopolgyClear(...) clear will clear any cable length values saved to flash memory, thereby

disabling the cable length check.

2.4 Increased Maximum Captures to 32 MPI 1261

In previous versions, the maximum number of controller captures was 16. This has been increased to 32.

2.5 Support for Multi-Vendor Flash Downloads MPI 1256

The MPI now supports the SynqNet node flash download for a number of different flash component vendors. This is where the sqNode FPGA image data is stored in flash memory on the node. These changes are transparent to the user. Flash components that are currently supported are:

Vendor	PN	Size	Circuit
Atmel	AT45DB021B	2M	Bowsprit
Atmel	AT25F2048N-10SI-2.7	2M	Bowsprit
SST*	SST25VF020-20-4C-SA	2M	Bowsprit
STM	M25P20-VMN6T	2M	Bowsprit
Xilinx	XC18V02	2M	Outrigger

2.6 ZMP/XMP Support Added to Device Driver MPI 1242

ZMP support has been added for the Windows NT, Windows 2000, and Windows XP device drivers. The new driver version is 3.00 and can be verified by running the VERSION.EXE utility. This driver is not compatible with earlier versions of the MPI. When upgrading MPI version you will need to upgrade device drivers. To insure that the new device is properly installed, be sure to install the new release using the installshield package from MEI.

2.7 SqNode Binary Download Improvements MPI 1219

In response to customer feedback in regards to sqNodeFlash, we have added some routines to the MPI to improve our sqNode download usability. The following routines have been added to the MPI:

```
meiSqNodeVerify(MEISqNode                  sqNode,
                MEISqNodeDownloadParams    *params);

meiSqNodeFpgaFileNameVerify(MEISqNode   sqNode,
                            char*       fileName);

meiSqNodeFpgaDefaultFileName(MEISqNode             sqNode,
                             MEISqNodeFileName     *fileName);
```

The sqNodeFlash utility uses the meiSqNodeFpgaFileNameVerify(...) routine to guarantee that the FPGA image being downloaded is compatible with the specified node. If the specified file image is not compatible, meiSqNodeFpgaDefaultFileName(...) is used to retrieve the default filename, which is displayed to the user. If the *-verify* flag is specified, meiSqNodeVerify(...) is used to verify that the correct flash image was downloaded correctly. The sqNodeFlash utility will also call meisqNodeInfo(...) to verify that the node type matches the image file.

2.8 Network Shutdown if all Nodes Fail MPI 1217

The SynqNet network will continue to operate in SYNQ (cyclic), even if one or more (but not all) nodes fail. Previously, if all the nodes failed, the controller would continue to operate the network in SYNQ mode. Now, if ALL the nodes fail, the controller will shutdown the network and the SynqNet state will change to DISCOVERY.

2.9 New Probe Object MPI 1216

A new object has been added to the MPI. The Probe object provides functionality similar to that of the capture module. The Probe object is less configurable than the capture module, but can be re-triggered several times in a single servo cycle. Unlike the capture module, there is no position computation in the firmware from Probe data. The Probe object can be used in "stand-alone" mode for position captures using only the MPI functions described here, or Probe can be used with the Data Recorder to provide time-based captures across several axes. There is a maximum of 256 probes per controller (4 per node, 32 motor type nodes, and 32 I/O type nodes).

Two sample applications using the Probe object can be found in the /apps folder.
probe1.c - Position captures using Probe in Time mode.
probe2.c - Position captures using Probe in Position mode.

The ***probe.h*** header file contains all of the methods and data types for the Probe object.

For more information about the data types and methods in the Probe object, please refer to the Software > MPI > Documentation section on MEI's Technical Support website.

2.10 New meiControlInfo(...) method MPI 1212

The MPI has been restructured so that the controller information is now located in the Control object. Use the meiControlInfo(...) method to retrieve the appropriate information about the controller.

```
long meiControlInfo (MPIControl      control,
                     MEIControlInfo   *info);

typedef struct MEIControlInfo {
      char    tNumber[MEIControlSTRING_MAX];
      char    serialNumber[MEIControlSTRING_MAX];
      char    boardType[MEIControlSTRING_MAX];
}MEIControlInfo;
```

Version 20031025

	New Version	Previous Version
Firmware	510A3	468A1
MPI Library	20031025	20030818

2.11 New Sample Rate Verification MPI 1246

The MPI now verifies that the controller sample rate is sufficient to accurately sample all Analog inputs (ADCs) on the network. An error will be returned if the controller sample rate configurations are not valid for ADC acquisition time.

2.12 Improved version checking of SynqNet node FPGA images MPI 1227

The version.exe utility now takes advantage of a new member of the MEISqNodeInfo{} structure and reports whether or not a SynqNet node is running with the "FPGA default" version. The default FPGA version is the recommended node FPGA image to use. The default version is compatible with all SynqNet nodes and has been thoroughly tested with the current version of the MPI.

2.13 Support for Soonhan 102K SynqNet Nodes MPI 1226

Software support has been added for the following Soonhan SynqNet nodes:

Model #	Description
IO2K-SynqNet-A-1	1 Axis Analog
IO2K-SynqNet-A-2	2 Axis Analog
IO2K-SynqNet-A-1A	1 Axis Analog with ADCs
IO2K-SynqNet-A-2A	2 Axis Analog with ADCs
IO2K-SynqNet-A-1S	1 Axis Analog with Aux DACs
IO2K-SynqNet-A-2S	2 Axis Analog with Aux DACs
IO2K-SynqNet-A-AS	1 Axis Analog with ADCs and Aux DACs
IO2K-SynqNet-A-2AS	2 Axis Analog with ADCs and Aux DACs
IO2K-SynqNet-P-1	1 Axis Pulse
IO2K-SynqNet-P-2	2 Axis Pulse

The node specific defines and methods are located in soonhan_io2k.h. Contact Soonhan Engineering for nodedetails: www. soonhan.co.kr/.

2.14 Support for AMC DQ Series Drives MPI 1221

Support has been added for the new AMC DQ Series Drives.

Amc_Digiflex DQ111EE
DQ111SE-H
DQ111SS
DQ111SE

The DQ-Series replaces the DRQ-Series. The DRQ-Series is obsolete and is no longer supported by the MPI library. Additional Error Messages have been added and the Amp Fault Mask has been changed to Amp Fault Codes. The I/O defines in amc_digiflex.h have been changed to support the DQ-Series I/O.

2.15 New Data Recorder Features MPI 1215

The number of available recorders has been increased from 1 to 32. The number of enabled recorders can be 0 to 32 and is specified by a new parameter in the MPIControlConfig{...} structure. The creation of a recorder object will require a second (new) parameter, *number*, in the mpiRecoderCreate(...) function.

Recorders are currently started only by the host and stop either on host command, or when a predetermined number of records have been stored. This start/stop functionality has not changed. However, new trigger control blocks within the recorders will allow the recorders to start or stop based on the states of internal variables. Each recorder will have two trigger blocks. Either block can start or stop the recorder. The start/stop triggers can be configured for either pattern matching or trigger-on-change. If additional trigger flexibility is needed, User Limits can be used in conjunction with the recorder triggers. The triggers are configured using a structure passed using the *external* pointer of mpiRecorderGet/Set().

When mpiRecorderStart(...) is called, recorder status is no longer cleared. The recorder status is now cleared when mpiRecorderConfigSet(...) is called with (config != NULL). The status clearing will return buffer pointers and counters to zero. mpiRecorderStart(...) can now resume recording where it was last stopped.

The following methods/data types were added/deleted/changed and can be found in ***recorder.h***.

mpiRecorderCreate(...) - The *number* parameter was added.
WARNING! This change can potentially break your application code.

mpiRecorderStart(...) - The *period* and *fullCount* parameters were removed.
WARNING! This change can potentially break your application code.

MPIRecorderConfig{...} - The *highCount* parameter was added.
mpiRecorderFlashConfigGet(...) - Deleted from recorder.h
mpiRecorderFlashConfigSet(...) - Deleted from recorder.h
MPIRecorderRECORDERS_MAX - Added to recorder.h
MPIRecorderMessage - Addition of MPIRecorderMessageNO_RECORDERS_AVAIL and MPIRecorderMessageNOT_ENABLED messages.

2.16 mpiAxisCommandPositionSet(...) Error Check MPI 1209

The mpiAxisCommandPositionSet(...) error check has been extended. If the controller is updating the axis's command position when mpiAxisCommandPositionSet(...) is called, MPIAxisMessageCOMMAND_NOT_SET will be returned. mpiAxisCommandPositionSet(...) now checks for the following conditions:

- Axis is in a STOPPING, STOPPED, or MOVING state.
- Any motor associated with the axis has the disableAction configuration set to MEIMotorDisableActionCMD_EQ_ACT and the motor's Amp Enable is disabled.
- If the command position read from the controller does not match the requested position.

2.17 Yaskawa Module Name Changes MPI 1199

The SqNodeLib modules for yaskawa have been renamed:

```
yaskawa_sigma3  -->  yaskawa_sgds
yaskawa_dmd     -->  yaskawa_sgdz_md
```

The datatypes, structures, and methods located inside the yaskawa header files have also been renamed to match the sgds and sgdz_md product names.

2.18 Topology Save/Clear Conflicts MPI 1192

There were a few conflicts between the TopologySave/Clear routines and the sqNode/Motor configurations that were resolved by adding the following rules to the meiSynqNetTopologySave/Clear routines:

- Topology cannot be cleared unless it has previously been saved.
- Object configuration items that use service commands cannot be saved to flash unless the topology has previously been saved.
- A Topology Save does not clear working memory configurations.
- Topology cannot be saved if it has previously been saved. (it must be cleared first)
- A Topology Clear will lose object configuration items that use service commands if the user has changed them from their default values.

2.19 Optimized Flash Memory MPI 837

In previous versions, saving settings to flash memory and downloading new flash images to the XMP over a client-server connection used to take a very long time. Changes have been made to optimize this process and now these processes take the same amount of time as if they were saved on a local XMP.

Version 20030818

	New Version	Previous Version
Firmware	468A1	459B1
MPI Library	20030818	20030620.1.1

2.20 Support for Trust SynqNet I/O Nodes — MPI 1196

Support has been added for the following Trust SynqNet I/O Nodes:

TA805-D TA806-D TA807-D
TA805-E TA806-E TA807-E
TA805-F TA806-F

2.21 Support for the Yaskawa SGDZ — MPI 1194

Support has been added for the following Yaskawa SGDZ models:

SGDZ-MD40EN7A
SGDZ-MD20EN7A

2.22 Pulse Support for SynqNet — MPI 1193

Pulse support has been added to the MPI. The following structures are used to configure the Pulse Output and reside in **stdmei.h**:

```
typedef enum MEIMotorStepperPulseType {
      MEIMotorStepperPulseTypeINVALID = -1,

      MEIMotorStepperPulseTypeSTEP,
      MEIMotorStepperPulseTypeDIR,

      MEIMotorStepperPulseTypeCW,
      MEIMotorStepperPulseTypeCCW,

      MEIMotorStepperPulseTypeQUADA,
      MEIMotorStepperPulseTypeQUADB,

      MEIMotorStepperPulseTypeLAST,
      MEIMotorStepperPulseTypeFIRST =  MEIMotorStepperPulseTypeINVALID + 1,

} MEIMotorStepperPulseType;

typedef struct MEIMotorStepperPulse {
      MEIMotorStepperPulseType   type;
      long                       invert;
} MEIMotorStepperPulse;

typedef struct MEIMotorStepper {
      long                       loopback;  /* TRUE = FPGA pulse feedback,
                                               FALSE = encoder feedback */
      MEIMotorStepperPulse       pulseA;
      MEIMotorStepperPulse       pulseB;
      double                     pulseWidth;/* output pulse width  (sec) */
```

```
        } MEIMotorStepper;
```

To use pulse outputs, you will need to:

1) Configure the motor type for STEPPER.
2) Set the motor's stepper pulse width. Make sure it meets the drive's requirements and is not smaller than 2x the minimum pulse period.
3) Enable the motor's stepper loopback if there is no encoder feedback.
4) Configure the motor's stepper pulseA and pulseB types for STEP, DIR, CW, CCW, QUADA, or QUADB.
5) Select the motor I/O's config type for pulseA and pulseB. This will route the pulseA/B signals to the node's digital outputs.

See the stepCfg1.c sample application for details.

2.23 mpiMotionAction(...) Optimization MPI 1191

In this version of the MPI, the method mpiMotionAction(...) was optimized for the MPIActionRESET action type. To make the execution as fast as possible, service commands to clear node status, CRCs, and the packet error counters were removed. Service commands to clear amp and encoder faults are only sent to the node when an amp fault or encoder fault is active.

A new method, meiSqNodeStatusClear(...) was added to clear node faults. It sends service commands to the node in order to clear the node status, CRCs, packet error counters, and hardware latches for the nodeDisable and powerFault.

2.24 SynqNet Node Analog Inputs MPI 1175

The RMB-10V, RMB-10V2 and some Trust nodes support analog inputs. MPI support has been added to support the reading of node-based analog inputs. The number of analog inputs a node supports can be determined with meiSqNodeInfo(...). An analog input value can be read with the method meiSqNodeAnalogIn. The analog to digital converted value is scaled from -1.0 to +1.0, where +1.0 is a full-scale positive voltage. The input range of the ADC is hardware-specific.

2.25 CD Drive Firmware 1.1.2 Supports Drive Warnings MPI 1155

Version 1.1.2 of the CD drive firmware supports drive warnings. These warnings are now displayed correctly in the sqDriveMsg.exe utility.

2.26 MPI DLL/Application Compatibility Checking MPI 1152

The MPI has version compatibility protection between the DLL and an application. This was added to protect customers who do NOT recompile their applications when upgrading to newer MPI software releases. If an application is NOT compiled with the same version as the MPI DLL, mpiControlInit(...) will return a "Control: Application not compatible with MPI DLL" message.

Although this protection is valuable, it also makes patch upgrades difficult. Whenever the MPI code is changed, the version is changed, and then ALL of the tools and applications must be recompiled. In reality, the tools and applications only need to be recompiled if and only if the MPI header files have been modified.

The compatibility protection feature was changed so that it only returns an error if there has been an interface change (header files do not match DLL).

The new algorithm uses an internal version number that gets incremented only when a critical change is made that breaks compatibility. For the MPI, a critical change is defined as an interface change. An internal version number has been added which defines the version of the last interface change.

For example:

```
    #define MPI_VERSION                  "20030620.1.1" // actual MPI version (exists in MPI today)
    #define MPI_INTERFACE_VERSION "20030620.1"    // new version that defines interface compatibility
```

The new internal version number (MPI_INTERFACE_VERSION) is now used in the existing compatibility check instead of MPI_VERSION. This will allow the MPI_VERSION to change as many times as needed, without affecting a compiled customer application.

3 Incremental Changes

Since the last Production Release of the MPI, version 20030620.1.1, API changes have been made to the header files that add features and fix various software and firmware bugs.   Please see the **\XMP\doc\header_diff.doc** file to see a compiled list of the incremental changes that have taken place between the following two versions of the MPI library.

New Version	Previous Version
20031222	20031025

4 Fixed Bugs of MPI/MEI Libraries:

Version 20031222

	New Version	Previous Version
Firmware	514A6	510A3
MPI Library	20031222	20031025

Unable to Clear Node Event Type *MPI 1272*

In MPI 20030620.1.9 and ealier, meiSqNodeEventReset(...) is unable to clear the node event type MEIEventTypeSQNODE_USER_FAULT. The definition for meiEventMaskSQNODE(mask) macro (stdmei.h) has been updated to correct the problem.

Amp Enable Bit in Wrong State *MPI 1271*

In MPI releases prior to 20031216, a bug existed where the Amp Enable bit on a node got stuck in the wrong state. This bug usually occurred after performing an mpiMotorFlashConfigSet(...) or when using Motion Console to save the Controller Summary to flash (with the "Include All Sub-objects" selected). This issue has been corrected.

Missing Home Event with Capture *MPI 1264*

In certain configurations, sometimes the home event was not triggered by a capture. The problem was caused by race condition between the firmware's foreground and background tasks. The capture state machine and status is updated in the foreground, the home limit and event status are updated in the background. If the foreground/background delta was large, the home limit would intermittently miss the capture. This problem was corrected by extending the capture status for one full background cycle, if the home limit is configured to trigger from the capture status and the home limit action is not "None."

Motion TIMEOUT Error *MPI 1259*

If a motor limit is exceeded and its conditions are met to generate a STOP action, then a call to mpiMotionStart(...) or mpiMotionModify(...) would return a TIMEOUT error. This would only occur if the motor limit conditions would continually generate the STOP action. It would not occur if the STOP was commanded by an application. This problem was caused by a previously added check for an axis error or stopped state inside the firmware, which caused the motion state machine to lock-up. This problem was corrected by removing the check.

Path Motion with Short Elements *MPI 1255*

In previous versions, if the path elements were closely spaced together compared to the velocity * time slice, the maximum velocities and accelerations would be exceeded. This was caused by a problem in the path points calculations in mpiPathMotionParamsGenerate(...). This problem was corrected by changing the point generation calculation algorithm.

TxTime = 95% Causes Network Initialization Failure *MPI 1252*

In previous releases, specific network topologies failed to initialize if the sample rate was 4000 and the TxTime was 95%. This problem was caused by bug in the SynqNet timing calculations. Also, in some cases the PLL would fail to lock. This problem was corrected with updated timing calculations and a slightly modified PLL in the Node FPGAs. See the FPGA release notes for more details about the PLL change. If the older FPGA (versions 301 and earlier) is used on a node, the MPI will return a MEISynqNetMessageSCHEDULING_ERROR if the timing values are in the range that would cause a PLL failure.

Version 20031025

	New Version	Previous Version
Firmware	510A3	468A1
MPI Library	20031025	20030818

Multi-threaded Performance Problems MPI 1251

In previous versions, a problem occurred when making MPI calls that take locks in a CRITICAL priority thread. The CRITICAL priority thread could be blocked for a very short time by lower priority threads that also make MPI calls that take locks. This problem has been corrected.

Motion Action Reset Unmaps Axis 0 MPI 1250

In previous versions, when mpiMotionAction(..., MEIActionRESET) was called with an unmapped motion object, Axis 0's Motion Supervisor pointer was set to NULL. If Axis 0 was executing a PT or PVT point motion, an E-Stop would occur and the axis would appear to run out of frames. This problem was caused by an internal Motion Supervisor -->Axis mapping routine. This problem has been corrected.

Random TIMEOUT Errors MPI 1240

In previous releases, occasionally there are random, MPIMessageTIMEOUT errors being returned from the MPI on specific PCs. Mostly, the errors occur from mpiControlReset(...), which internally calls meiControlSampleWait(...). The failure was caused by a bug in certain chipsets that cause the Win32 QueryPerformanceCounter(...) to randomly jump ahead by 4 to 5 seconds. The MPI uses this counter to check how much time has elapsed when waiting for a specific number of controller samples. For more details, see the report from Microsoft at http://support.microsoft.com/default.aspx?scid=kb;en-us;274323.
This problem was corrected by using a lower resolution millisecond "tick" timer from the host.

meiEventMaskALL(...) Missing Event Types MPI 1238

In previous versions, meiEventMaskALL(...) did not set the SynqNet, SqNode, or Can event masks. This has been corrected.

Encoder Reversal Configuration Problem MPI 1236

In previous versions, saving the network topology to flash caused the encoder reversal configuration to be disabled. This problem was corrected.

Application DLL Compatibility Check MPI 1229

In previous releases, if a return value of MPIControlMessageLIBRARY_VERSION was returned from mpiControlInit(...) the SynqNet network would not be initialized. This has been changed so the network will still be initialized even in the event of this warning.

Capture Configuration Get MPI 1228

In previous releases, if mpiCaptureConfigGet(...) was called on a capture object that had been previously configured, the configuration information would not be correct. The mpiCaptureConfigGet(...) might have read the configuration information from the wrong node. This problem was corrected.

Multi-Point Motion Problem MPI 1223

In previous releases, if a PVT (or other multi-point motion) move was stopped and then an SCurve (or other point-to-point) move was executed, the motion supervisor could enter into an ERROR state. The problem was caused by the point buffer's low and empty limits not being disabled by the mpiMotionStart/Modify(...) for the second move, which triggered an E-Stop. This problem has been corrected.

CANOpen 20k Bit Rate Problem MPI 1220

In previous CAN firmware versions, the CANOpen interface did not work correctly at a bit rate of 20k. All other bit rates worked correctly. This problem was corrected with CAN firmware version CAN002A6.out.

CANOpen Heartbeat Problem MPI 1218

In previous CAN firmware versions, the XMP did not generate its heartbeat message and did not correctly monitor the health of a node that used the heartbeat. This problem did not affect nodes that use node guarding to maintain their health. This problem was corrected in CAN firmware version CAN002A6.out.

mpiAxisCommandPositionSet(...) with fractional values MPI 1068

In previous versions, if mpiAxisCommandPositionSet(...) was called with a fractional command position, it would return a MPIMessageCOMMAND_NOT_SET code. This problem was caused by the comparison check between the requested command position (double) and the controller's command position (long). This problem has been corrected.

Version 20030818

	New Version	Previous Version
Firmware	468A1	459B1
MPI Library	20030818	20030620.1.1

Network Initialization Lockup MPI 1198

In previous versions, meiSynqNetInit(...) would not return if the network was shutdown and all the nodes were powered off or disconnected from the controller. This problem was caused by an optimization routine that was used to detect if any nodes were found during network initialization. The optimization logic was corrected in version 20030818.

Service Command Busy Loop MPI 1185

A tight, busy loop in the MPI service command routine was causing process starvation in real-time systems. Although this bug is hard to reproduce, it can be experienced in a multi-threaded application when CPU bandwidth is critical (such as in an application that streams points from the host to the controller). To correct this problem, the busy loop was opened to allow threads of the same priority to take control of the CPU. This problem was corrected in version 20030818.

Service Command Busy Loop MPI 1184

If a CANOpen node supports more than 64 bits of digital input/output, or more than 4 channels of analog input/output, the CAN interface generates messages of the wrong length. This problem was corrected in version 20030818.

PVT Stairstep velocity profile bug — MPI 1170

In previous versions, a bug existed when running PVT or PVTF motion profiles. This bug caused the acceleration values for the moves to be zero, which created a "stair-step" velocity motion profile. This problem was corrected in version 20030818.

Default Configuration for Encoder Faults — MPI 1169

The default configuration for Encoder Faults used to be Primary and Secondary when a secondary encoder was available. The default configuration for Encoder Faults is now Primary only.

Unable to Invert Aux Encoders — MPI 1165

A bug existed where the MEIXmpMotorConfig structure in the firmware contained an incorrect address for Aux Encoders on all nodes except the first node. As a result, these nodes were unable to change the default configuration for secondary encoders. This bug was corrected in 20030818.

mpiCaptureConfigGet(...) method returns MPIMessageUNSUPPORTED — MPI 1164

A bug existed in the mpiCaptureConfigGet(...) method where capture objects that were configured for any captureMotorNumber that did not have a secondary encoder mapped to it would return MPIMessageUNSUPPORTED. This problem was corrected in version 20030818.

sqCmd.exe Supports Downstream Data Field — MPI 1160

When sqCmd.exe was used to read some direct commands to a drive, the downstream data could not be specified and always defaulted to zero. The sqCmd.exe utility has been modified and now uses the /data command line argument to specify the downstream data field. This problem was corrected in version 20030818.

Incomplete error messages in message.exe utility — MPI 1156

In previous versions, some of the return codes returned by the MPI did not have any strings associated with them. The message.exe utility gave no indication as to the nature of the error. This problem was corrected in version 20030818.

Modifications to sqDriveMonitor.exe Utility — MPI 1154

When using the /poll option of the sqDriveMonitor.exe Utility with certain drives, Windows would generate an illegal memory access message box. This problem was corrected in version 20030818.

Setting MPIControlConfig.axisCount to Zero Causes an MPI Library Assertion — MPI 1150

In previous versions, setting a value of zero for the control config axisCount would cause an MPI library assertion error. This problem was corrected in version 20030818.

5 Existing Bugs of MPI/MEI Libraries:

Unexpected Error returned when using meiSynqNetPacketConfigSet(...) MPI 1180

If all members of the MEISynqNetPacketCfgMotor{} structure are set to NONE or Zero (0x0), the MPI should effectively disable the motor on the network and cause the controller to renumber the subsequent motors on the network. Instead, the following error will be returned: "ERROR 0x191d: SynqNet: invalid encoder count." This is a bug and will be corrected in future releases.

mpiControlConfigSet(...) overwrites Custom Packet Configuration MPI 1179

A bug exists in the mpiControlConfigSet(...) method when it is called after the meiSynqNetPacketConfigSet(...) method. Because there are conditions that require the mpiControlConfigSet routine to reinitialize the SynqNet network, this bug can reset any previously set custom packet configurations back to the default packet configuration. This bug can be avoided by performing mpiControlConfigSet(...) method calls before using any meiSynqNetPacketConfigSet(...) method calls. This is a bug and will be corrected in future releases.

Error with the SynqNet Packet ioInput Configuration MPI 1178

A bug exists in the meiSynqNetPacketConfigSet(...) routine that causes an internal packet data misalignment when the packet payload for an MEI_RMB node is configured with an ioInput value of MEISynqNetResourceIoBitsNONE. The effects of this data misalignment are currently unknown and could potentially cause a return value of "0x1904: SynqNet: network communication is down." However, this bug is not considered to be a safety concern.

Paired with the ioInput and ioOutput packet data fields, there are 16bits of dedicated I/O packet data for each motor. Since the dedicated I/O are fixed fields, removing the general purpose ioInput and ioOuput will not gain any network bandwidth.

It is recommended that the ioInput and ioOutput packet configuration fields remain unchanged when performing a meiSynqNetPacketConfigSet(...)

Vague Error Code returned when accessing drive parameters MPI 1159

When performing a drive parameter read/write to a particular parameter that is unsupported by a drive, the following incorrect error message is returned: “ERROR 0x1c05: sqDispatch: Node specific command dispatch error.” The correct return value should be a “not supported” error.

Win2000 Device Driver System Stand by Error MPI 741

The XMP Windows 2000 device driver will not allow a host system go into "Standby" or "Hibernation" mode. This bug will be corrected in a subsequent release.

6 Motion Console and Motion Scope

6.1 Closed Issues: Motion Console

	New Version	Previous Version
Motion Console	03.40.05	03.40.04

Modified in Version: 03.40.05

Modification Type: ***DR (Discrepancy Report)***

Number **Name**

1011 General Purpose Motor I/O Is Broken For Any Node Other Than The First One

In the SqNodeLib, a node has a table of I/O descriptions for each motor on the node. Motion Console accesses this table using the motor number. It should be subtracting the nodes motor offset from the motor number.

	New Version	Previous Version
Motion Console	03.40.04	03.40.01

Modified in Version: 03.40.04

Modification Type: ***DR (Discrepancy Report)***

Number **Name**

1011 General Purpose Motor I/O Is Broken For Any Node Other Than The First One

In the SqNodeLib, a node has a table of I/O descriptions for each motor on the node. Motion Console accesses this table using the motor number. It should be subtracting the nodes motor offset from the motor number.

Modified in Version: 03.40.03

Modification Type: ***NF (New Feature)***

Number **Name**

1003 [Dup. of 1001] New Save Topology to Flash Features

Changes made to the MPI regarding saving and clearing topology have prompted the following changes to Motion Console:

1) Save Topology button is disabled if topology is already saved
2) Clear Topology button is disabled if topology is not saved
3) The user is prompted to save topology to flash if they try to save an object to flash that requires the topology to be saved.

Modification Type: ***DR (Discrepancy Report)***

Number **Name**

992 No Warning of Topology Mismatch After Network is Reinitialized

The user normally sees a warning if an attempt is made to save topology to flash when the topology does not match either volatile or non-volatile memory. This warning is not being displayed if the network is shutdown, the topology is altered, and the network is brought up again.

1000 [Dup. of 998] Motor I/O labels are not updated when changing topologies

Under certain situations, the Motor I/O labels will not reflect the correct labels. This occurs when:

1) Motion Console is minimized.
2) A SynqNet node at a particular position is swapped out with a node that has different I/O labels.
3) Motion Console is restored.

Modified in Version: 03.40.02

Modification Type: ***NF (New Feature)***

Number **Name**

971 Add ability to display sqNode I/O

The I/O Summary can now be configured to display SqNode I/O.

987 [Dup. of 980] Add Clear Status Button to SqNode Summary

Add a button to the SqNode Summary, which will call meiSqNodeStatusClear() for the selected node when it is clicked.

996 [Dup. of 984] Make Multiple Recorders Accessible

The new multiple recorder feature in the MPI has been made accessible in Motion Console.

*Modification Type:****MI (Minor Improvement)***

Number **Name**

986 [Dup. of 982] Error from mpiAxisCommandPositionGet() when downloading firmware

When firmware is downloaded to a controller, all motion supervisors that are mapped to an axis will report the following MPI error:

> "mpiAxisCommandPositionGet: Argument invalid"

Modification Type: ***FE (Future Enhancement)***

Number **Name**

977 Add Secondary Encoder Index to Motor I/O Status Window

Display the secondary encoder index in the I/O status window of the Motor Summary.

Modification Type: ***DR (Discrepancy Report)***

Number **Name**

985 [Dup. of 983] Download Firmware Confirmation Dialog Box shows up on taskbar

The dialog box that pops up for user confirmation before downloading firmware shows up on the Windows taskbar. This means that the user is NOT locked out of Motion Console. Any action can be made in Motion Console while the dialog box is open.

988 [Dup. of 979] Binary download button should be disabled on a failed node

The "Binary Download" button should be disabled if the node has failed.

989 [Dup. of 976] Toolbar in Motion Console can be lost

If the toolbar on the main frame is dragged off so that it is floating, and then closed, there is no way to get it to reappear without restarting the application. Furthermore, the state and position of the floating toolbar is not saved, so it cannot be restored properly when the application starts up.

Modification Type: ***CR (Change Request)***

Number **Name**

978 **Separate I/O Into Groups**

In the I/O Summary, draw a line that separates the I/O associated with different motors.

	New Version	Previous Version
Motion Console	03.40.01	03.39.07

Modified in Version: 03.40.01

Modification Type: ***NF (New Feature)***

Number **Name**

972 **Add support for stepper motors**

Added support for stepper motors to the Motor Summary window.

Modification Type: ***MI (Minor Improvement)***

Number **Name**

974 **Add support for MEIMotorFaultBitFEEDBACK_FAULT_SECONDARY**

MEIMotorFaultBitFEEDBACK_FAULT_SECONDARY was added to the MPI, so it was also added to the Motor Summary window.

Modified in Version: 03.39.08

Modification Type: ***FE (Future Enhancement)***

Number **Name**

962 **Panic button should default to ABORT**

The Motion Supervisor panic action, the action taken when the panic button is clicked, now defaults to ABORT as opposed to NONE.

Modification Type: ***DR (Discrepancy Report)***

Number **Name**

964 **Ctrl + click on object icon doesn't open a minimized window**

Clicking on the object icon in the main toolbar while holding Ctrl + click will now show the summary window in a normal state if it was previously iconized.

Modified in Version: 03.39.07

Modification Type: ***CR (Change Request)***

Number **Name**

959 **MPIMotionTypeP and MPIMotionTypePF removed from MPI**

The MPIMotionTypeP and MPIMotionTypePF motion types were removed from the MPI and so were also removed from MotionConsole.

6.2 Closed Issues: Motion Scope

	New Version	Previous Version
Motion Scope	01.21.11	01.21.09

Modified in Version: 01.21.10

Modification Type: ***DR (Discrepancy Report)***

Number	Name

916 Erroneous error when importing data

Importing files with banding enabled now works.

922 Cannot open .PAN when banding was enabled

Importing of .PAN files which had been saved with banding enabled now works.

	New Version	Previous Version
Motion Scope	01.21.09	01.21.06

Modified in Version: 01.21.09

Modification Type: ***DR (Discrepancy Report)***

Number	Name

1010 User masking disabled for 16-bit types

User bit-masking is now possible for the new 16-bit data types.

Modified in Version: 01.21.08

Modification Type: ***DR (Discrepancy Report)***

Number	Name

994 File Open re-applies mask to already-masked user-defined data

File Open (of a *.pan file) disables masking and shifting attributes (if they are enabled) for user-defined traces (and the stock 16-bit "types" that are masked and shifted internally) so that the data is read in as-is without further masking and shifting disabled. This is ok as this Pane will be read-only and the mask and shift will not be applied to any new input.

1009 Motion Scope Velocity Units Problem

The command and actual velocity units for the displayed and exported data were in units of counts/sample. These are now in the correct units of counts/second.

	New Version	Previous Version
Motion Scope	01.21.06	01.21.06

Modified in Version: 01.21.06

Modification Type: ***CR (Change Request)***

Number	Name
951	**Build Motion Scope Release configuration with MPI Release configuration**

Prior to this change, the Release configuration of Motion Scope was linked to the Debug configuration of the MPI. The Release configuration of Motion Scope is now linked to the Release configuration of the MPI.

6.3 Open Issues: Motion Console

Issue Type: ***CR (Change Request)***

Number **Name**

991 Add an Info Tab to the Motor Summary

A tab needs to be added to the Motor Summary to display data in the MEIMotorInfo structure.

Issue Type: ***DR (Discrepancy Report)***

Number **Name**

393 CellTips don't work for checkboxes

If the text of a cell does not fit within the cell of an Object Attribute Grid, then the CellTip should display the complete text of the cell. This feature doesnot work for cells containing checkboxes.

427 Grid Not Always Drawn Correctly When Selection Changes

Sometimes, selected cells are not being drawn as selected (i.e. with the colors inverted) until some window event occurs. One way to reproduce this bug is to select the entire table by clicking on the top, leftmost cell of the grid. When this is done, some cells in the grid are sometimes not drawn as inverted, but then drawn correctly when the user clicks on the grid or hovers over a control, causing a tooltip to be displayed.

561 Last column cannot be sized to the edge of the grid

The width of the last grid column cannot be moved to the edge of the grid. If the vertical scroll bar is present, then attempting to resize the last column will cause the width to snap to a distance of 4 pixels to the left of the right edge of the grid.

569 Gray button drawn in origin cell when 1st column is minimized

A faulty button is drawn in the origin cell when the following procedure is followed:

1) select the entire first column of the Motion Supervisor Actions tab grid;
2) slide the column width to the narrowest possible width. This results in the gray button appearing to be a combination of all the buttons in the column.

628 Horizontal Scroll Bar Behaves Strangely When Large Numbers of Objects are Displayed

When some summary windows are programmed to display a large number of objects (more than 20), the scroll bar will behave strangely.

637 Creative position zero behavior

If the controller is in open loop sine comm mode, the command position doesn't zero when the "Zero Position" button in the Motion Supervisor summary is clicked unless the "Clear Fault" button is clicked first.

657 "(Not Available)" listed as an option in pulldown menu

In the Motor Summary window, under the I/O configuration tab, all XCVR Config pull-down menus list "(Not Available)" as an option.

741 User In bit not reported

The User In bit is not reported when the bit is toggled.

761 Pulldown boxes only work on primary monitor with a multiple monitor setup on win2k

When using Motion Console on a win2k system with multiple monitors, the pull down boxes don't function on the secondary monitor, but function on the primary monitor.

847 Flickering could appear on several status windows

Some flickering could appear on the Axis, Motion, and Motor Status pages because of a bug that affects how event status flags are compared.

887 Object settings not saved prior to opening a new .INI file

Changes to object attributes that are stored in the .INI file are not saved when another .INI file is opened or created.

913 Column width is forgotten after MoCon restart

When a column width has been modified by the user, the new width should be associated with the object and saved in the .INI file. That way the column width can be restored to the users preference when the summary window is

914 Motion Console crashes on phase 2 to phase 1 conversion

Motion Console for the SynqNet Phase 1 branch (MPI 20011220) is not forward compatible with the .INI file created by Motion Console for the SynqNet Phase 2 branch (MPI 20021212 and on). The .INI file should be removed when reverting from the newer version back to the older version.

949 Motion Console hogs CPU when finding errors.

In some situations, Motion Console will use 100% of the CPU. This occurs when the MPI is waiting for a command to timeout. Usually a particular type of object is causing the problem, so closing the summary for that window will greatly improve response time by the computer.

954 Scaling in Summary Windows Partially Implemented

The size of the grids in a Summary window can be scaled using Ctrl+MouseWheel. The following needs to happen for the feature to be fully implemented:

1) Scale all grids in the summary window, not just the one with the focus
2) Adjust max size of Summary window after scaling
3) Adjust max splitter position after scaling
4) Provide some more obvious method of setting the scale

963 Tab does not change focus cell when editing number

970 Motion Console incorrectly labels Ka2

Motion Console incorrectly labels PIV filter parameter Ka2 (the one after Ka1) as N/A and will not let the user enter a value for it. It should be accesable by the user.

997 Refresh data before saving to flash

1006 [Dup. of 992] No Warning of Topology Mismatch After Network is Reinitialized

The user normally sees a warning if an attempt is made to save topology to flash when the topology does not match either volatile or non-volatile memory. This warning is not being displayed if the network is shutdown, the topology altered, and the network brought up again.

1007 Motion Console error is sometimes incorrect

1008 Topology warning is misleading

1019 Motion Console may not show CAN node live status correctly

1021 A change from string to ring after a controller reset is not shown in the SynqNet Summary window.

Issue Type: ***MI (Minor Improvement)***

Number **Name**

739 Add more detail to tooltips for disabled controller buttons

When a button on the Controller Summary is disabled because the controller is not initialized, some clues can be added to help the user rectify the situation.

740 Add greater detail to toolbar button tooltips concerning various modes of operation

The action that is executed when a toolbar button is clicked can sometimes be modified by holding down the Ctrl or Shift keys. The nature of this modified behavior should be described in detail in the tooltip for each button.

775 Remove Broken Wire and Illegal State for Motor I/O page

Broken Wire and Illegal State statuses are displayed both in the Motor I/O and Status tab pages. They are somewhat inappropriate for the I/O window because there are no I/O pins associated with them.

975 Library function errors in mocon at newtork shutdown

6.4 Open Issues: Motion Scope

Issue Type: ***CR (Change Request)***

Number	Name
1018	**No Recorders Available**

Issue Type: ***DR (Discrepancy Report)***

Number **Name**

542 **Motion Scope fails to draw data on Windows 98**

With triggering set to "Go Button" and "Stop Button," data will accumulate (as seen by the XOffset value changing), but no traces will be drawn. Changing the status of "View/Status Bar" will cause the pane to draw the traces. This problem occurs frequently, but irregularly. We have not found a way to reliably reproduce the problem. We have also not seen this problem on Windows NT.

643 **Odd behavior when opening a .pan file**

Here are the steps to reproduce the bug:

1. Open up a .pan file (previously created with File Save from Motion Scope).
2. Immediately hit the go button.
3. While the plots are being generated, right-click somewhere on the pane and the graphing will mysteriously disappear.

Now, if you use the Stop button to halt data acquisition, click "Traces" to bring up the Traces list dialog and then hit the “OK” button, the problem will be solved and any graphing you do after this will not have this behavior.

679 **Ctrl-LMB value display hides Y-units label.**

Pane Export not supporting "hex" display format.

713 **Motion Scope Data not aligned with scale lines**

When collecting/displaying data, sometimes the data points don't align properly with the scale markers on the X axis. This is easiest to see by turning on the "sample band" in the Pane Display configuration and Displaying in Units of Samples. The problem can be corrected by forcing a re-draw of the data: sliding the data on/off the screen, minimizing/maximizing, or zooming in/out.

769 **Motion Scope hangs when opening file multiple times**

Motion Scope will sometimes hang when opening a .PAN file. This can be recreated by opening a .PAN file and then closing the pane. Repeat until the hang occurs: usually after the 4th or 5th time.

776 **AutoScale occasionally fails to utilize last portion of data in Range for selected Trace.**

AutoScale occasionally fails to utilize last portion of data in Range for selected Trace.

781 **Motion Scope displayes graph as if it missed a sample when it really didn't**

While using Motion Scope to record the sample counter while I was testing motion modify code. Motion Scope displayed some data as if it missed a sample, but while investigating the sample counter I see that Motion Scope really didn't. Perhaps there is some rounding error in the calculation of elapsed time during the motion?

806 **Motion Scope looses all traces after a SynqNet node dissapears**

When a SynqNet node disappears when using Motion Scope, Motion Scope displays some error messages and then the pane being used vanishes. This can be reproduced by plotting some information with Motion Scope and then pulling the SynqNet cable on the first node. This can be particularly troublesome if special traces have been set up and not saved. It is easy to remove a pane but it takes a lot of work to recover settings if they were needed.

852 **Time scale on Motion Scope does not refresh upon sample rate change**

When the sample rate on the XMP is changed, Motion Scope is not aware of it.

877 **Shift key inhibits dragging of YRangeBar slider edge**

When the cursor is placed on the YRangeBar slider edge and a shift-drag is attempted, and the tooltip window is open, then the tooltip window is dragged instead of the slider edge. Without the shift key down, the tooltip window closes automatically and the drag works fine.

896 **[From MPI Libraries and Firmware :] moScope "Save Pane" Corrupts timebase.**

The following problem was reported using MPI 20021219:

1. Open Motion Scope and configure it to record a measured position, then capture some data.
2. Save the Motion Scope pane using File -> Save As.
3. Close the view using File -> Close.
4. Load the previously saved pane using File -> Open.

1016 Motion Scope hangs when zooming upon read-only pane files

1017 Strange gaps in data when using Go/Stop buttons

Issue Type: ***MI (Minor Improvement)***

Number	**Name**
473	**Dialog boxes missing ToolTips**

None of the dialog boxes display ToolTips.

662 Parameter precision (number of digits to right of decimal point) for X and Y axis labels.

Add parameters that provide the ability to modify the precision (number of digits to right of decimal point) for X and Y-axis data labels. Add a separate parameter for the X-axis and parameters per Trace on the Y-axis.

663 Groups to be supported in File Import input FFT files.

Groups to be supported in File Import input FFT files.