Motion Engineering, Inc.

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

APPLICATION NOTE 9D00-0130, REV. A

MPI/XMP Data Tracing

INTRODUCTION

This application note describes the usage of MPI/XMP Data Tracing. The purpose of the MPI/XMP data tracing is to provide visibility into the data being used within the MPI/XMP motion system. This visibility allows for easy trouble shooting with minimal impact to overall system performance. This additional data tracing is applied using the current trace module interface. Although data tracing is very similar to traditional MPI tracing, data tracing focuses more upon the values (data) that the MPI is carrying/computing than upon the operation of the MPI.

DATA TRACE INTERFACE

Additional Trace information

Three new object trace bits have been added to support data logging:

MEIControlTraceGATE_DATA	← Traces control gate sets and resets
MEIMotionTracePARAM_DATA	← Traces MPI motion parameters sent to XMP
MEIEventMgrTraceEVENT_DATA	← Traces XMP events

These bits must be "turned on" for each object that is desired to generate the data trace information. To do this, use **mei*Object*TraceSet(object, traceMask)**. When these bits are turned on, multiple trace macros in each corresponding module are evaluated.

Document Revision History: Application Note 9D00-0130				
Rev.	**Date**	**Description**	**DCR No.**	**DCO No.**
➡ A	15MAR2001	Released.	533	1324

TRACE DATA SYNTAX

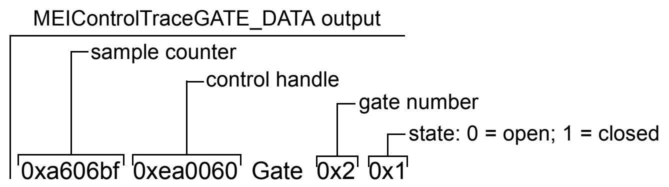
The trace data syntax changes slightly with every different object and motion type. However, all trace statements begin with the XMP sample counter and the Control handle address. The idea behind the trace output format is to: 1) output only the needed information; and, 2) add just enough text to be human readable without the need for post-processing.

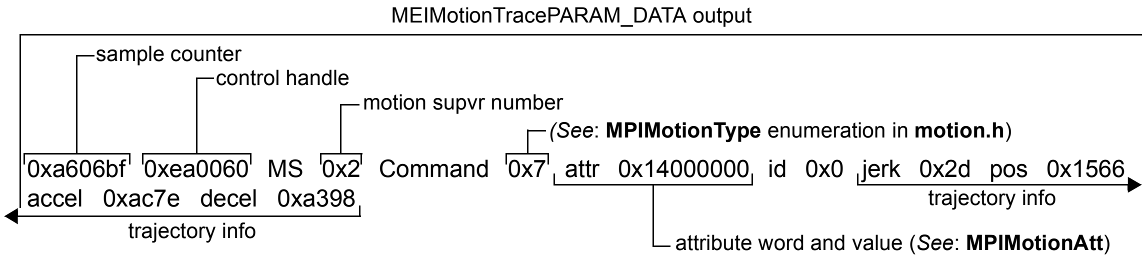
Here is a sample trace output from a motion application with all three bits turned on:

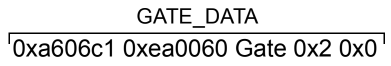
Sample trace output:

```
0xa606bf 0xea0060 Gate 0x2 0x1
0xa606bf 0xea0060 MS 0x2 Command 0x7 attr 0x14000000 id 0x0 jerk 0x2d pos 0x1566
accel 0xac7e decel 0xa398
0xa606c1 0xea0060 Gate 0x2 0x0
…
0xa60de6 0xea0060 Motion 0x2 Event 0x9 src 0xc8ad18 Data 0xa60de6 0x0 0x1564
0x1566 0xa60de6 0xa60de6 0xa60de6 0xa60de6 0xa60de6 0xa60de6
```

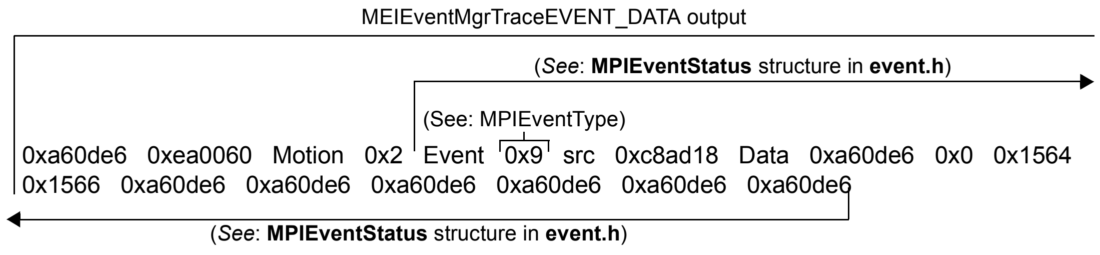
Sample trace data explained:


MEIControlTraceGATE_DATA output
sample counter
control handle
gate number
state: 0 = open; 1 = closed
0xa606bf 0xea0060 Gate 0x2 0x1


MEIMotionTracePARAM_DATA output
sample counter
control handle
motion supvr number
(*See*: **MPIMotionType** enumeration in **motion.h**)
0xa606bf 0xea0060 MS 0x2 Command 0x7 attr 0x14000000 id 0x0 jerk 0x2d pos 0x1566
accel 0xac7e decel 0xa398
trajectory info
trajectory info
attribute word and value (*See*: **MPIMotionAtt**)



...


MEIEventMgrTraceEVENT_DATA output
(*See*: **MPIEventStatus** structure in **event.h**)
(See: MPIEventType)
0xa60de6 0xea0060 Motion 0x2 Event 0x9 src 0xc8ad18 Data 0xa60de6 0x0 0x1564
0x1566 0xa60de6 0xa60de6 0xa60de6 0xa60de6 0xa60de6 0xa60de6
(*See*: **MPIEventStatus** structure in **event.h**)

Motion Gate Trace Data

The syntax of this data is the sample count, control handle and address, gate number, and closed data.The first and fourth in from the above example shows a gate set and a gate reset, respectively.

Motion Parameter Trace Data

The syntax of this data is based upon the motion type and the attributes set for the move.

The second line in the above sample shows an S-Curve move on motion supervisor #2. This move has the moveId attribute bit set with an ID = 0. The rest of the line shows the trajectory and position information for the move.

Event Trace Data

Syntax of this data is based upon the event type.

The fifth line in the above example shows a MOTION_DONE event from motion supervisor #2. The application was set up so the event data word[1] would be the moveId, word[2] would be actual position and word[3] is command position. The rest of the data words are not configured for use (but still reported). The MPI allows the application to set up the ten data words to capture XMP values from anywhere on the controller at the time when the event occurs.