MTM Net

Machine Tool Measurement Network

Contents

[Introduction 4](#_Toc120274851)

[MTM Net Functions 5](#_Toc120274852)

[MTM Net Servers 6](#_Toc120274853)

[NC Gage & HMI 6](#_Toc120274854)

[Activating MTM Net 6](#_Toc120274855)

[Viewing Connected Clients 7](#_Toc120274856)

[MTM Net Functions 8](#_Toc120274857)

[NC Server 10](#_Toc120274858)

[Activating MTM Net 10](#_Toc120274859)

[Viewing Connected Clients 10](#_Toc120274860)

[MTM Net Functions 10](#_Toc120274861)

[Example Clients 11](#_Toc120274862)

[NC Server 11](#_Toc120274863)

[Configuring an MTM Net connection in NC Server 11](#_Toc120274864)

[MTM Net services consumed by client 12](#_Toc120274865)

[Machine Status 12](#_Toc120274866)

[NC Gage Status 12](#_Toc120274867)

[NC Gage Reports 14](#_Toc120274868)

[Data Exchange 14](#_Toc120274869)

[Macro Upload 15](#_Toc120274870)

[MTM Net Test Tool 16](#_Toc120274871)

[Using the test tool 16](#_Toc120274872)

[Connect to NC Gage 17](#_Toc120274873)

[Accessing Macro Variables 17](#_Toc120274874)

[Read Variable 17](#_Toc120274875)

[Write Variable 17](#_Toc120274876)

[Accessing Tool Information 17](#_Toc120274877)

[Getting a list of tools 17](#_Toc120274878)

[Getting a tool offset value 17](#_Toc120274879)

[Setting a tool offset 18](#_Toc120274880)

[Accessing Work Offset Information 18](#_Toc120274881)

[Getting the current active work offset 18](#_Toc120274882)

[Getting the values of a work offset 18](#_Toc120274883)

[Updating values of a work offset 18](#_Toc120274884)

[Appendix A: Function Use 19](#_Toc120274885)

[Appendix B: Proto File 20](#_Toc120274886)

# Introduction

MTM Net is a gRPC-based inter-process networking technology used to communicate with MTM software products. MTM Net is the basis of MTM product APIs. The MTM Net is used to enable status messaging as well as data transfer between products.

The following MTM Software products implement MTM Net Servers:

* NC Gage
* Smooth OMM
* NC Server

The following MTM Software products implement MTM Net clients:

* NC Server
* NC Measure

In addition, other Hexagon products implement MTM Net clients:

* IMC
* SMIRT
* EDGECAM

For more information about gRPC please see the following resources online:

* Official gRPC web site: [gRPC.io](https://grpc.io/)
* Microsoft Learn: [gRPC | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/architecture/cloud-native/grpc)

Note: If working in Visual Studio and using .Net gRPC support is available as of .Net 4.6

For other operating system or alternative implementations, visit the gRPC Github site: [grpc · GitHub](https://github.com/grpc)

## MTM Net Functions

The following functions are currently available over MTM Net:



# MTM Net Servers

## NC Gage & HMI

### Activating MTM Net

The MTM Net server in NC Gage is configured using the NC Gage Configuration Tool on the “Advanced” tab. The NC Gage Configuration Tool is located in the NC Gage installation directory (typically: C:\Program Files (x86)\Metrology\NC Gage v4.2)

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

To enable the MTM Net Server, check the Enable MTM Service checkbox

If the PC or controller running NC Gage has more than one IP Address, select the one to be used with the MTM Net server. By selecting “Any” all network interfaces on the PC are enabled for MTM Net.

(Optional) If you would like to enable transmitting DFQ files to the NC Server, check the Enable DFQ Transmission checkbox.

Save the configuration, Exit the Config tool and start NC Gage.

### Viewing Connected Clients

The NC Gage Connected Probing Dashboard displays the number of MTM Net connected clients.

Graphical user interface, application

Description automatically generated

For details about the connected clients, navigate in NC Gage settings and then to the Information Manager

|  |  |
| --- | --- |
|  |  |
| Logo, company name  Description automatically generated | Click the Gear icon (settings) |
| Logo  Description automatically generated with medium confidence | Click the “>>” button to page over in the menu |
| A picture containing logo  Description automatically generated | Click the “!” button (Information Manager) to open the information dialog |
| Icon  Description automatically generated | Click the MTM Net icon to view connected clients |

This displays the IP Address of any connected clients and the connection time:Graphical user interface, text, application

Description automatically generated

### MTM Net Functions

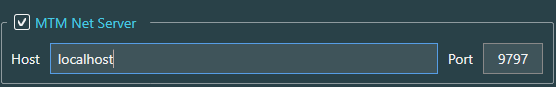
The MTM Net server in NC Gage & HMI provides the following functions:

|  |  |
| --- | --- |
| **Function** | **Description** |
| Get Machine Status | Returns the machine status as one of the following:   * Offline * Idle * Running * Alarm |
| Get NC Gage Program Status | Returns the program status as one of the following:   * None * Executing * Execution Succeeded * Execution Error * In DCM Mode * Program Name * Error Info |
| Upload File | Uploads a file to the Machine. This can be one of the following:   * Macro * Report * XMDF File |
| Get Journals | Sends all available journal files to the caller |
| Get DFQ Files | Sends all available DFQ files to the caller |
| Read Macro Variable | Returns the value of a macro variable |
| Write Macro Variable | Writes a value to a macro variable |
| Get CNC Type | Returns the controller type |
| Get CNC Model | Returns the controller model |
| Get Version | Returns the following version information   * MTM Net Version * NC Gage Version * CNC API Version |
| Get Axis Info | Returns the following information about the axes configured in NC Gage:   * Axis Name * Kinematic Type * Axis Number |
| Get Report File List | Returns a list of all available measurement reports |
| Download Report File | Returns the requested measurement report |
| Get Tool Number | Returns the number of tools |
| Get Tool Offset | Returns the tool offset values for the specified tool |
| Set Tool Offset | Sets the tool offset values for the specified tool on the control |
| Get Machine Configuration | Returns the following information about the machine configuration´   * Base variable address * Settings variable address * Program base address * Move speed * Rotary speed * Probing speed * Units * Rotary axis center points * Rotary axis vectors |
| Request Interactive | Requests one of the following actions:   * Update Work Offset * Update Tool Offset * Write Variable * Read Variable |
| Get Interactive Result | Returns the result of the requested interactive action |
| Get PRG Previews | Gets a thumbnail preview for a PRG file from the NC Server |
| Read Position Stream | NOT Implemented yet |
| Read Position | Returns the current position of the machine in machine coordinates |
| Read Position in Current Work Offset | Returns the current position of the machine in the active work offset coordinates |
| Get Active Work Offset | Returns the currently selected work offset |
| Get Calibration Result | Gets latest rotary table calibration data |

## NC Server

### Activating MTM Net

The MTM Net server can be enabled or disable in the NC Server settings dialog:



To activate the server, click the checkbox next to MTM Net Server.

By default, the server will be bound to the local host, meaning that clients will be able to contact the server by attempting to connect to the IP address associated with the local host PC.

If the PC has more than one IP address exposed to the network, you may set this to 0.0.0.0. This will bind the service to all IP addresses exposed to the network.

If the IT administrator would like to limit the server to being available only at a specific address, the address should be entered here.

The default port to connect with the service is 9797. If the PC is protected by a firewall, you must ensure that traffic on this port is allowed, or change the port number in NC Server to be a port that is already available on the firewall (ie 80, or 8080).

### Viewing Connected Clients

It is not possible currently to view the connected client information

### MTM Net Functions

The MTM Net Server in NC Server provides the following functions:

|  |  |
| --- | --- |
| **Function** | **Description** |
| Get Machine List | Returns a list of all machines configured in NC Server, including the following information:   * Machine Name * Machine ID * Machine Controller Type |
| Upload Inspection Plan File | Uploads an Inspection plan file and creates a PC-DMIS program for a specific machine. (Requires an existing Template.PRG for that machine) |
| Upload CAD File | Uploads a CAD file and imports it into the active PC-DMIS Program |

# Example Clients

## NC Server

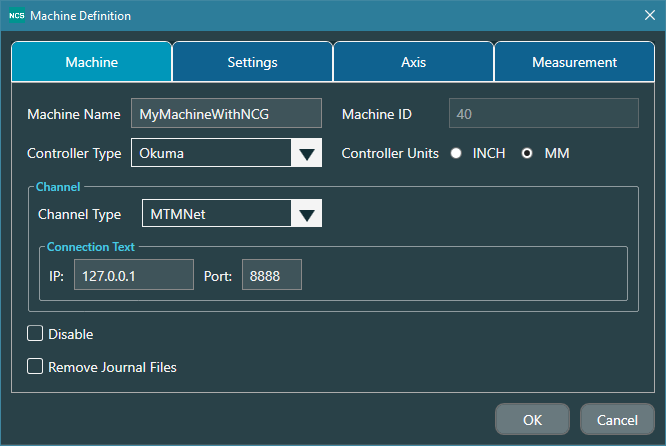
The NC Server supports using the MTM Net link to NC Gage as a substitute for the traditional file sharing mechanism that was used in the past.

This offers the user the benefits:

* Simplified setup and connection
* More information in real-time about the machine
* More data exchange between systems
* Link to NC Gage data as well as PC-DMIS Data

### Configuring an MTM Net connection in NC Server

To configure the MTM Net connection for a machine in NC Server, open the machine definition dialog and select the Tab “Machine”.



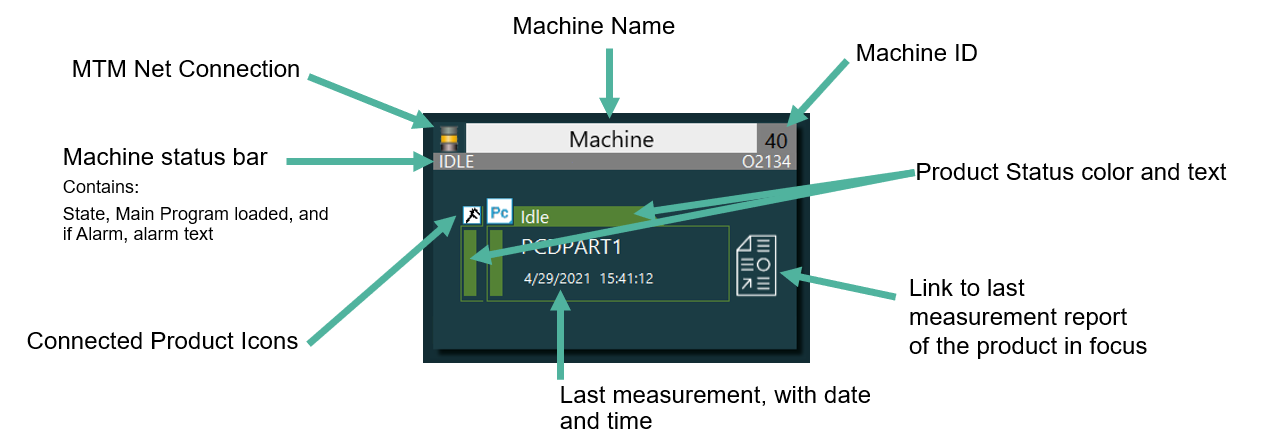
Select the Channel Type “MTMNet” form the channel type dropdown.

NOTE: MTM Net is currently available for Okuma and Mazak only.

Enter the IP address and port of the NC Gage system (shown in the NC Gage Config Tool on the machine).

Close the dialog (after all other settings have been checked)

The machine card for the machine will have two tabs on it and the MTM Status Light Tree indicating the machine status:



### MTM Net services consumed by client

The NC Server uses many of the services provided by MTM Net Server in NC Gage.

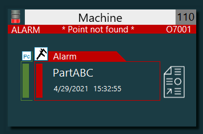
#### Machine Status

The machine status is displayed on the light tree icon, and the Machine Status Bar.

The light tree icon will change colors (Red, Yellow, Green or Grey) following the same rule as the NC Gage Light Tree on the NC Gage Connected Probing Dashboard. (Note: A small delay in updates from NC Gage to NC Server is possible due to network and software processing).

The Machine Status Bar will contain Name of the current state, under the light tree, and the program number of the current program loaded on the control. The bar remains grey all the time except if in the “Alarm” state. If the machine is in the “Alarm” state, the Machine Status Bar will be in red, and the error message associated with the alarm will be displayed on the Machine Status Bar.

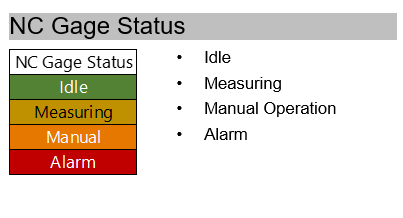
*Example of an alarm during an NC Gage measurement program*



#### NC Gage Status

The status of NC Gage is displayed on the colored tabbed area for the NC Product.

The colored areas of the NC Gage Product Tab will change colors depending on the state of NC Gage as follows:



IDLE – NC Gage is in DCM Mode and ready to run a measurement program

Measuring – NC Gage is running a program

Manual Operation – NC Gage is being operated by the operator (NC Gage is NOT in DCM Mode)

Alarm – An NC Gage Alarm has been raised

Offline (Not shown) – NC Gage is not running, no connection to the MTMNet

##### Examples:

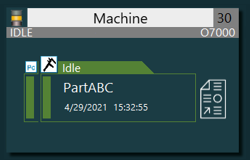
|  |  |
| --- | --- |
|  |  |
| Machine offline | Machine Idle, NC Gage is in DCM Mode |
|  |  |
| NC Gage Alarm. | Machine is running, NC Gage is not in DCM |

#### NC Gage Reports

When NC Gage creates a new report, this report should be available at the NC Sever after completion of the NC Gage program.

##### Last report

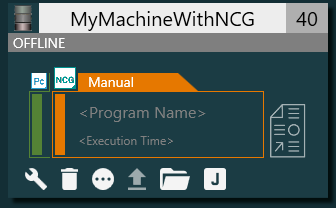
Once the Measurement status has changed back to idle, clicking the report icon next to the NC Gage Product Tab should open the NC Gage report for the program and time displayed on the tab.



Pressing the Report Icon should open the report for NC Gage program “PartABC” run at 15:32 on April 29

##### All reports

All NC Gage reports can be accessed by ensuring the NC Gage product tab is selected, then clicking the folder icon on the floating toolbar



Clicking the Folder Icon opens a dialog listing all reports from NC Gage

You may then select the report you would like to view and open it.

#### Data Exchange

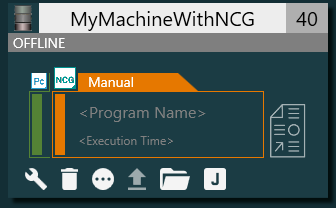
In addition to the various status information supplied by MTM Net, data exchange is also used by NC Server to support measurement and measurement feedback for PC-DMIS NC.

##### Data Exchange Functions

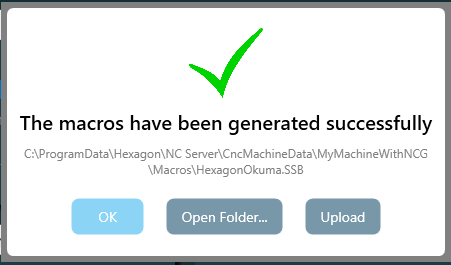
|  |  |
| --- | --- |
| Journal File Transfer | When using MTM Net, Journal files are not shared with NC Server via a folder. The NC Collect function built into NC Gage collects the measuring data for PC-DMIS NC, then transmits the Journal data over the MTM Net when the journal is created. |
| DFQ File transfer | If enabled on the NC Gage side, DFQ files will be automatically transferred to the NC Server and stored under the respective machine directory. |
| PC-DMIS NC Update Work Offset  PC-DMIS NC Update Tool Offset  PC-DMIS NC Write Variable  PC-DMIS NC Read Variable | The REQ file created by PC-DMIS is transferred to the NC Collect function in NC Gage via the MTM Net. The NC Collect function then processes the REQ as usual. |

#### Macro Upload

When a machine has been configured in NC Server, macros for use with PC-DMIS NC can be generated by pressing the “Generate” button on the floating toolbar.



If the machine is online, when the macros are generated, you will have the option in the dialog to upload them:



Clicking “Upload” will send the generated macros to the machine via MTM Net.

Macros can also be uploaded to the machine by using the upload button on the floating toolbar.



Uploading a CNC program is also available via MTM Net.

## MTM Net Test Tool

An example application has been developed that can be used to test many of the MTM Net functions available from NC Gage. Parties interested in implementing MTM Net in their own applications can get the example project by contacting the MTM team. The example project is a C# WPF desktop application.

### Using the test tool

The test tool can be used by either downloading a Zip file that contains the executable file and its dependencies, or by downloading the Visual Studio project and building it on a developer PC.

The tool can be downloaded here:

[https://downloads.ms.hexagonmi.com/PC-DMIS-NcGage/Release/MTM\_NET\_Test\_Tool](https://eur02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdownloads.ms.hexagonmi.com%2FPC-DMIS-NcGage%2FRelease%2FMTM_NET_Test_Tool&data=05%7C01%7C%7C98f8e0a492714e341b1008db7173f8d9%7C1b16ab3eb8f64fe39f3e2db7fe549f6a%7C0%7C0%7C638228515156170578%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=FxtUosh3ulLxU6%2BpYt8Fe7wpIhJM85CycTX%2F9fOUsUs%3D&reserved=0)

**NOTE**: If you download the tool, please check the properties on “DemoWpfClient.exe” that it is not blocked by the operating system because of it being downloaded from the internet.

Graphical user interface, text, application, email

Description automatically generated

The application is launched by running the application “DemoWpfClient.exe” which will open a window as shown here:

Graphical user interface

Description automatically generated

To use the test tool you will need to have NC Gage installed either on a supported machine tool in the local network, on the local PC or running somewhere in the local network.

#### Connect to NC Gage

Enter the IP Address of the device where NC Gage is running. Enter a port for establishing the MTM Net connection. The default port is 8510. This port number must match the port number defined in the NC Gage MTM Net settings in the NC Gage configuration tool.

Click the Connect button, and the notification box should show that the connection is successful.

### Accessing Macro Variables

Macro Variables on the control can be written to and read out. These operations can be used from the Machine Operations tab.

#### Read Variable

To read a variable from the machine, enter the variable number in the text box labled “Macro Address” then click <Read Macro> .

The result will appear in the textbox “Macro Value”

#### Write Variable

To write to a variable on the machine, enter the variable number in the text box labled “Macro Address”, and the value to be written in the textbox “Macro Value”. Click <Write Macro> and the value will be written to the machine.

### Accessing Tool Information

Use the the tab “Tool Info” to work with tool information on machining centers.

Graphical user interface

Description automatically generated

#### Getting a list of tools

The list of available tool numbers can be retrieved from the machine by clicking <Get Tool Number>

The result list will appear in the list box to the right of the button

#### Getting a tool offset value

To get the value of a specific tool offset, enter the appropriate values specifying which value to get, then click <Get Tool Offset>.

The result will be shown in the text box labeled “Tool Offset Value”

#### Setting a tool offset

To set the value of a specific tool offset, enter the appropriate values specifying which value to set. Enter the value to be written in the text box labeled “Tool Offset Value” then click <Set Tool Offset>

### Accessing Work Offset Information

Select the Tab “Work Offset” to work with work offset information:

Graphical user interface, table

Description automatically generated with medium confidence

#### Getting the current active work offset

To determine which work offset is active on the machine, click <GetActiveWorkOffset>.

The name of the active work offset (ex. “G58”) will appear in the textbox labeled “Workoffset Number”

#### Getting the values of a work offset

To read all values stored in a work offset on the machine, enter the work offset number in the text box “Workoffset Number” (ex. “G55”), then click <Get Work Offset>

The offset values will appear in the text boxes associated with each axis as labeled.

#### Updating values of a work offset

To modify values of a work offset on the machine, enter the work offset number to be modified in the text box “Workoffset Number”. Then enter the values for each axis to be updated. Click <UpdateWorkOffset> and the new values will be written to the work offset on the machine

# Appendix A: Function Use

The table outlines which MTM Net functions provided by NC Gage are used by other products.

# Appendix B: Proto File

The Protocol buffer definition file (proto file) is defined as follows. (This is an example only, for the current version please contact MTM).

syntax = "proto3";

import "google/protobuf/timestamp.proto";

import "google/protobuf/empty.proto";

package Hexagon.MTM.V1;

service MTMService {

  rpc GetMachineStatus(google.protobuf.Empty) returns (stream GetMachineStatusResponse); //Server response stream

  rpc GetProgramStatus(google.protobuf.Empty) returns (stream GetProgramStatusResponse); //Server response stream

  rpc UploadFile(stream UploadFileRequest) returns (UploadFileResponse); //Client request stream

  rpc GetJournals(stream GetJournalsRequest) returns (stream GetJournalsResponse); //Both ways stream

  rpc GetDFQFiles(stream GetDFQFilesRequest) returns (stream GetDFQFilesResponse); //Both ways stream

  rpc GetReportFileList(google.protobuf.Empty) returns (GetReportFileListResponse);

  rpc DownloadReportFile(DownloadReportFileRequest) returns (stream DownloadReportFileResponse); //Server response stream

  rpc RequestInteractive(InteractiveRequest) returns (InteractiveResponse);

  rpc GetInteractiveResult(InteractiveResultRequest) returns (InteractiveResultResponse);

  rpc GetPRGPreviews(stream PRGPreviewsRequest) returns (stream PRGPreviewsResopnse); //Both ways stream

  rpc GetMachineConfiguration(google.protobuf.Empty) returns (GetMachineConfigurationResponse);

  rpc GetCalibrationResult(google.protobuf.Empty) returns (GetCalibrationResultResponse);

}

enum MachineStatus {

  machine\_offline = 0;

  machine\_idle = 1;

  machine\_running = 2;

  machine\_alarm = 3;

}

enum ProgramStatus {

  prog\_none = 0;

  prog\_executing = 1;

  prog\_exec\_success = 2;

  prog\_exec\_error = 3;

}

enum UploadFileTypes {

  cnc\_macro = 0;

  cnc\_measurement\_routine = 1;

  pcd\_report = 2;

}

enum InteractiveTypes {

  update\_workoffset = 0;

  update\_tooloffset = 1;

  write\_var = 2;

  read\_var = 3;

}

enum ImageTypes {

  png = 0;

  jpeg = 1;

  bmp = 2;

}

enum KinematicsTypes {

  Linear = 0;

  RotaryTable = 1;

  Wrist = 2;

}

message GetMachineStatusResponse {

  MachineStatus State = 1;

  string AlarmInfo = 2;

  string ProgramNumber = 3;

}

message GetProgramStatusResponse {

  ProgramStatus State = 1;

  bool IsDCM = 2;

  string ProgramName = 3;

  string ExecErrInfo = 4;

}

message UploadFileRequest {

  bytes FileData = 1;

  string FileName = 2;

  bool FileEOF = 3;

  bool NeedUploadMacro = 4;

  UploadFileTypes FileType = 5;

}

message UploadFileResponse {

  bool IsSuccess = 1;

  string ErrInfo = 2;

}

message GetJournalsRequest {

  int64 FileId = 1;

  bool FileOK = 2;

}

message GetJournalsResponse {

  int64 FileId = 1;

  string FileName = 2;

  bytes FileData = 3;

  bool FileEOF = 4;

}

message GetDFQFilesRequest {

  int64 FileId = 1;

  bool FileOK = 2;

}

message GetDFQFilesResponse {

  int64 FileId = 1;

  string FileName = 2;

  bytes FileData = 3;

  bool FileEOF = 4;

}

message ReportFileEntity {

  string FileName = 1;

  int64 FileSize = 2;

  google.protobuf.Timestamp LastWriteTimeUTC = 3;

}

message GetReportFileListResponse {

  repeated ReportFileEntity ReportFileList = 1;

}

message DownloadReportFileRequest {

  string FileName = 1;

}

message DownloadReportFileResponse {

  bytes FileData = 1;

  bool FileEOF = 2;

}

message InteractiveRequest {

  InteractiveTypes ReqType = 1;

  string ReqName = 2;

  bytes ReqData = 3;

}

message InteractiveResponse {

  bool IsSuccess = 1;

  string ErrInfo = 2;

}

message InteractiveResultRequest {

  InteractiveTypes ReqType = 1;

  string ReqName = 2;

  bool NeedSave = 3;

}

message InteractiveResultResponse {

  bool IsSuccess = 1;

  string ErrInfo = 2;

  double ValueOfReadVar = 3;

}

message PRGPreviewsRequest {

  string PrgFileName = 1;

  ImageTypes ImageType = 2;

  bytes ImageData = 3;

  bool ImageDataEOF = 4;

}

message  PRGPreviewsResopnse {

  string PrgFileName = 1;

  ImageTypes ImageType = 2;

}

message GetMachineConfigurationResponse {

  bool IsSuccess = 1;

  string ErrInfo = 2;

  string FileContent = 3;

}

message CalibratedAxis {

  string AxisName = 1;

  KinematicsTypes KinematicsType = 2;

  double X = 3;

  double Y = 4;

  double Z = 5;

  double I = 6;

  double J = 7;

  double K = 8;

}

message GetCalibrationResultResponse {

  bool IsSuccess = 1;

  string ErrInfo = 2;

  string FileContent = 3;

  repeated CalibratedAxis CalibratedRotaryAxes = 4;

}