

CeramillMotion Remote Control

Programming Guide

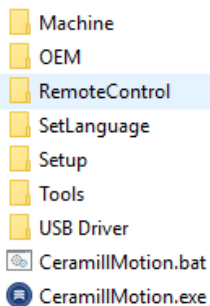
Author: Günter Marte
Date: 2018-06-13
Software: EST1274-49.00 or later
Document: EDA0021-05

1. Introduction

The CeramillMotion software provides 2 different interfaces to remotely control the software running on a PC and a connected milling/grinding machine.

Your Application can take control over the CeramillMotion system in 2 manners:

1. **Communication via a WCF service** (Windows Communication Foundation). This is the convenient method if your application is .NET based.
2. **File I/O based communication.** A defined folder is used where you write a file containing command and arguments. CeramillMotion generates subsequently a result file which you can evaluate. This interface is intended to be used by applications not based on Microsoft .NET.



Both interfaces are easy to implement and the file I/O method is nearly as fast as the WCF approach. Since the data volume transferred between your application and CeramillMotion is very small, it's just a choice of taste which one to use.

In the CeramillMotion installation path (mostly something like C:\Program Files\Ceramill Software\Machine) you can find all useful information and tools. Your interest should be focused on the Folder RemoteControl including this documentation, examples, test applications and configuration data.

2. Starting CeramillMotion

If you are using the WCF method, you must start CeramillMotion.exe at least once with administrator privileges to register the required URL for the communication service. In a typical customer scenario, this is done with our setup installation.

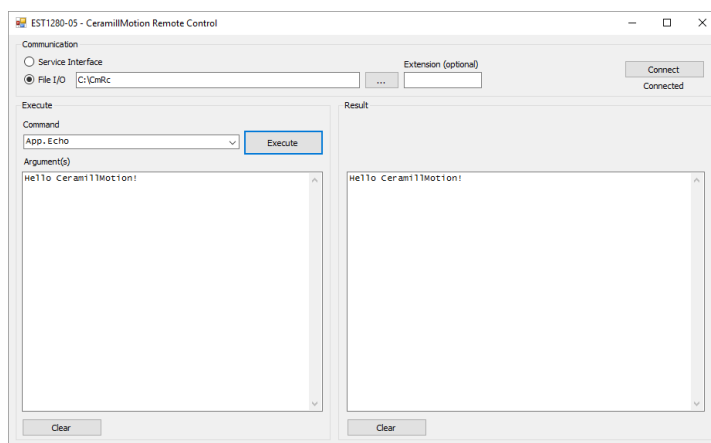
Normally CeramillMotion starts with a user interface, but probably you want to see nothing and let the software "work behind". You can do this by calling:

```
CeramillMotion -hide (hide an show nothing)
CeramillMotion -hide -NotifyIcon (hide an show a notify icon in the taskbar)
CeramillMotion -hide -NotifyIconShowOnClick (as above + show CeramillMotion on click)
```

You can also make CeramillMotion visible/invisible at any time with appropriate commands.

3. First steps

Run the test application CeramillMotionRemoteControl.exe located in subfolder \Test. Try it and play with it.



Execute different commands (using the drop down list or type it in) and look at the results. Note: some commands requires one or more arguments.

You will discover that the communication between this tool and CeramillMotion is text based and very simple.

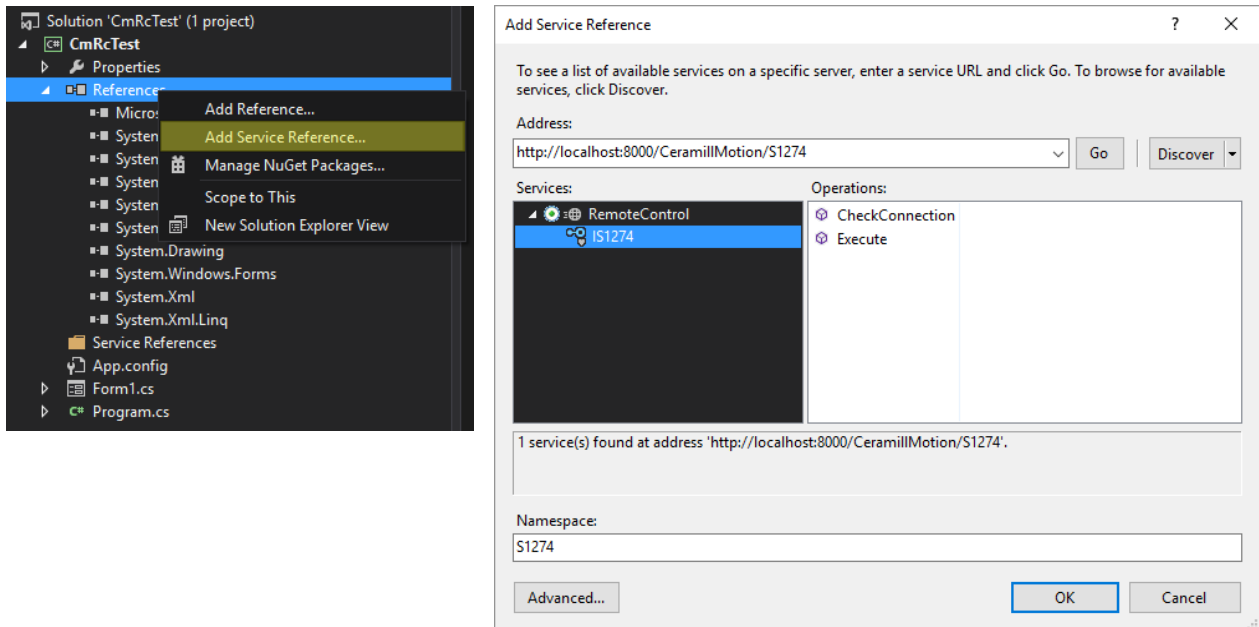
Dive deeper and observe the source code located in \Examples\EST1280-05.

4. Implement the communication

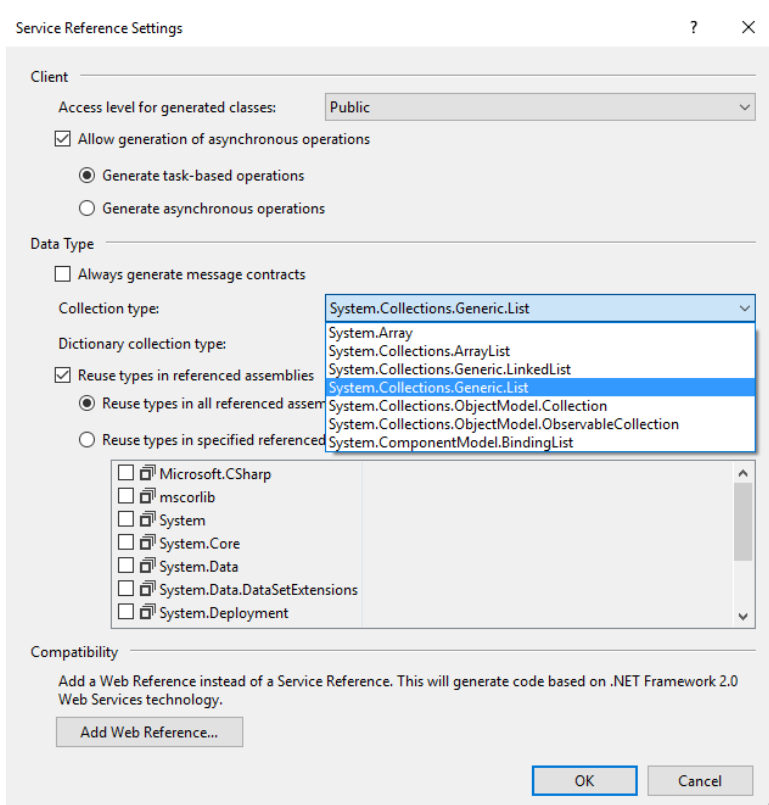
4.1. WCF

For this method your .NET-based application has create a service reference to

`http://localhost:8000/CeramillMotion/S1274`



Use Namespace `S1274` and set Collection type to `Systems.Collections.GenericList` in the Advanced Settings.



Add a `using` statement to `YourApplicationNamespace.S1274` (in this example `CmRcTest.S1274`)

```
using System;
...
using CmRcTest.S1274;

namespace CmRcTest
{
    ...
}
```

The Service S1274 provides only 2 methods:

```
void CheckConnection();

List<string> Execute (string Command, List<string> Arguments);
```

Now you can code something like this:

```
private void button1_Click(object sender, EventArgs e)
{
    var RemoteControl = new S1274Client();

    try
    {
        RemoteControl.CheckConnection();
    }
    catch (Exception ex)
    {
        MessageBox.Show (ex.Message);
        return;
    }

    List<string> Arguments = new List<string>();
    Arguments.Add ("Hello");
    Arguments.Add ("CeramillMotion");
    Arguments.Add ("I will control you!");

    List<string> Result = RemoteControl.Execute ("App.Echo", Arguments);

    foreach (string r in Result)
    {
        MessageBox.Show (r);
    }
}
```

If necessary, you can change the communication settings in `CeramillMotion.exe.config`:

```
<baseAddresses>
  <add baseAddress="http://localhost:8000/CeramillMotion/S1274"/>
</baseAddresses>
```

4.2. File I/O

To do this, CeramillMotion and your application shares a common folder, defined in `\Config\EST1274.RemoteControl.config`. This is the default:

```
RC.FileIoFolder = C:\CmRc
```

CeramillMotion creates this folder on startup and deletes the folder on closing.

To execute a command, you have to write a text file named `execute` (with no extension) into this folder. CeramillMotion is polling the common folder and executes your command if this file exists. After execution, CeramillMotion creates a file `result` and deletes `execute`. You can now check and delete `result`.

format of file execute: `command;argument_1;argument_2;...;argument_n`

Example: `App.Echo;hello;World`

Accepted separators are “;” and Tabulator

Format of file result: `result_1;result_2;...;result_n`

Example: `hello;World`

Feature added in EST1274-49.12:

You can optionally add an extension for the execute and result files - e. g. `execute-123` and `result-123`

This is useful if two or more remote controllers are involved to control the CeramillMotion application.

The first remote controller can write/read the files `execute1` and `result1`, the second one uses `execute2` and `result2`...

5. Commands and Results

Commands are case-insensitive, so you could use for example `app.echo` as well.

Some results are divided into “Sub-Results”, like

```
Error|E|0490|Machine|No compressed air  
Stop|S|0362|Machine|Protection shield open
```

on command `Status.GetMessages`.

Commands are categorized in the Groups App, Status, Resource, Machine, Job, CfgPar.

Commands of a Group are separated with “.” like `Machine.Connect`.



AMANNGIRRBACH

6. Commands

6.1. Command Overview

Command	Min. Version	Description
App.		
GetSoftwareId	EST1274-49.00	Get software version information of CeramillMotion.exe
Echo	EST1274-49.00	Returns the arguments
Hide	EST1274-49.00	Hide CeramillMotion (make invisible) Optional arguments (EST1274-49.13 or newer): NotifyIcon displays an icon in the status area of the taskbar NotifyIconShowOnClick displays an icon in the status area of the taskbar. CeramillMotion will be shown if user clicks the icon.
Show	EST1274-49.00	Show CeramillMotion (make visible)
IsHidden	EST1274-49.13	Returns True if CeramillMotion is hidden, otherwise False
Close	EST1274-49.00	Close CeramillMotion
UserCloseDisable	EST1274-49.13	Prevents the user from closing CeramillMotion
UserCloseEnable	EST1274-49.13	Enable application closing by the user
GetLanguages	EST1274-49.00	Get available languages
SetLanguage	EST1274-49.00	Set language (Ceramill Motion UI and messages), Example: EN
Status.		
GetMessages	EST1274-49.00	Get active messages (errors, stop conditions, warnings, informations)
Refresh	EST1274-49.00	Reset Errors(s) if possible and refresh Messages
IsAnyErrorActive	EST1274-49.00	True, if any error is active
IsAnyErrorStopActive	EST1274-49.00	True, if any error or stop condition is active
IsAnyErrorStopWarningActive	EST1274-49.00	True, if any error or stop condition or warning is active
Resource.		
GetMachinesKnown	EST1274-49.00	List all machines which has ever been connected to this computer
GetMachinesOnline	EST1274-49.00	List all machines which are currently connected to this computer
UpdateMachinesOnline	EST1274-49.00	Update (re-search) machines
SelectMachine	EST1274-49.00	Select the machine to use subsequently
GetSelectedMachine	EST1274-49.00	Get the current selected machine
Machine.		
Connect	EST1274-49.00	Connect to the machine selected with Resource.SelectMachine
Disconnect	EST1274-49.09	Disconnect machine (use this command before connecting to another machine)
IsConnected	EST1274-49.00	True, if machine is connected
ToggleRun	EST1274-49.00	If the machine is running, stop it and vice versa
IsRunning	EST1274-49.00	True, if machine is running (executes job)
Job.		
IsReadyToLoad	EST1274-49.00	True, if the machine is ready to receive a new job (no job in progress or job finished)
Load	EST1274-49.00	Load a new job (full path to the file as argument)
LoadAbort	EST1274-49.13	Aborts an active Load process
LoadGetStatusAndProgress	EST1274-49.13	Gets status and progress of the currently running load process. Result is: Status Progress – Example: JobTransferToMachine 75 Status can be: Idle Abort WaitForUserAction MoveToBlankPlacementPosition JobInfo JobAnalyze JobCalcExecutionTime JobTransferToMachine Finalize
Get	EST1274-49.03	Get the filename(s) of the currently loaded Job. No Result if no job is loaded or loading a job is in progress
Reset	EST1274-49.02	Resets the current loaded job (rewind to start, reset execution time)



AMANN GIRRBACH

Command	Min. Version	Description
Delete	EST1274-49.02	Deletes the current loaded job
GetProgress	EST1274-49.02	Get Job Progress (in %)
GetTimeSpanExecuted	EST1274-49.02	Executed time since job start, example: 02:23:17 or 8597 with argument Seconds
GetTimeSpanRemaining	EST1274-49.02	Remaining time, example: 01:57:33 or 7053 with argument Seconds
GetDateTimeReady	EST1274-49.02	Expected date and time for job completion, example: 13.05.2017 02:11:19
Maintenance.		Command Group Maintenance
Hourmeter	EST1274-49.12	Hourmeter of the machine
NextMachineMaintenance	EST1274-49.12	Next machine maintenance is required if hourmeter reaches this limit
NextSpindleCleaning	EST1274-49.12	Next spindle cleaning (in hours)
NextExhaustionMaintenance	EST1274-49.12	Next exhaustion maintenance (in hours)
NextCoolantCheck	EST1274-49.12	Next coolant check (date)
NextCoolantCleaning	EST1274-49.12	Next coolant circulation cleaning (date)
CfgPar.		Command Group CfgPar (Machine Configuration and Parameter)
GetMachineType	EST1274-49.00	Get the machine type
GetSerialNumber	EST1274-49.00	Get the serial number
GetWetProcessingAvailable	EST1274-49.00	True, if the machine is capable of wet processing
GetAxisCount	EST1274-49.00	Get number of axis (4 or 5)
GetValue	EST1274-49.00	Get value of a specified machine parameter

For commands in the Group CfgPar, the machine has to be selected but an active connection is not required. Information is determined from the pool of known machines if the machine is not connected.

6.2. Command Return Values

Command	Predefined Return Values											Return Values, Comments, Example
	OK	True/False	Missing Argument	Invalid Argument	No Result	Machine not known	Machine not selected	Machine not online	Machine not connected	File not found	Not allowed	
App.GetSoftwareId												EST1274-49.02 1274 49 2
App.Echo												
App.Hide	x			x								
App.Show	x											
App.IsHidden		x										
App.Close	x											
App.GetLanguages												List of all supported languages, EN English...
App.SetLanguage	x		x	x								
Status.GetMessages					x							No Result if no message is available
Status.Refresh	x											
Status.IsAnyErrorActive		x										
Status.IsAnyErrorStopActive		x										
Status.IsAnyErrorStopWarningActive		x										
Resource.GetMachinesKnown					x							



AMANN GIRRBACH

Command	Predefined Return Values											Return Values, Comments, Example
	OK	True/False	Missing Argument	Invalid Argument	No Result	Machine not known	Machine not selected	Machine not online	Machine not connected	File not found	Not allowed	
Resource.GetMachinesOnline												EST1274-49.13 or newer: returns <i>Busy</i> while searching for machines
Resource.UpdateMachinesOnline	x										x	Not allowed if a machine is connected
Resource.SelectMachine			x	x		x						
Resource.GetSelectedMachine							x					
Machine.Connect	x						x	x				
Machine.Disconnect	x							x				
Machine.IsConnected		x					x					
Machine.ToggleRun	x						x		x			
Machine.IsRunning		x					x		x			
Job.IsReadyToLoad		x							x			
Job.Get					x							
Job.Load	x		x						x	x	x	Not allowed if machine is running. Check <i>Machine.IsRunning</i> before executing this commands.
Job.Reset	x								x		x	
Job.Delete	x								x		x	
Job.LoadAbort	x											
Job.LoadGetStatusAndProgress												Status and Progress while loading a job
Job.GetProgress					x				x			Progress in % (0..100)
Job.GetTimeSpanExecuted					x				x			Format of date and time is based on the regional settings of the PC running CeramillMotion. Use argument <i>Seconds</i> to get timespans in total seconds. No <i>Result</i> if no job loaded.
Job.GetTimeSpanRemaining					x				x			
Job.GetDateTimeReady					x				x			
Maintenance.Hourmeter									x			
Maintenance.NextMachineMaintenance									x			
Maintenance.NextSpindleCleaning									x			
Maintenance.NextExhaustionMaintenance									x			
Maintenance.NextCoolantCheck									x			
Maintenance.NextCoolantCleaning									x			
CfgPar.GetMachineType					x		x					
CfgPar.GetSerialNumber					x		x					
CfgPar.GetWetProcessingAvailable		x			x		x					
CfgPar.GetAxisCount					x		x					
CfgPar.GetValue		x	x	x	x		x					Value format depends on type (integer, float, bool...)