

Application Notes

UDEMemtool Automation Basics

UDE is based on **Microsoft COM Technology**. Each client application that can use COM components (**ActiveX controls**) is able to remote-control UDEMemtool. This includes various scripting hosts.

This application note is mainly an annotated script written with Microsoft JScript to be executed on Windows Scripting Host. It shows how to load UDEMemtool component, connect target and load and flash-program a hex file.

Starting point

Here we assume you already know how to manually start UDEMemtool.exe, select and connect a target system and load and flash-program a hex file.

First step is to create an instance of COM class "UDE.Memtool":

```
var ProgId = "UDE.Memtool";  
var Memtool = new ActiveXObject(ProgId);
```

We have to prepare some paths and files. Adapt this to meet your environment:

```
var WorkDir = fso.GetAbsolutePathName(".");  
var CfgFile = WorkDir + "\\TriBoard_TC27x_dap.cfg";  
var HexFile = WorkDir + "\\MulticoreDemo.hex";  
var LogFile = WorkDir + "\\Memtool.log";
```

To diagnose problems we switch on logging-to-file and set the log level:

```
Memtool.CreateLogFile(LogFile, 50);
```

Now we initialize the component by using a target configuration file:

```
Memtool.Init(CfgFile);
```

Connect the target system:

```
Memtool.ConnectTarget();
```

Load the target application file:

```
Memtool.OpenFile(HexFile, true);
```

Program all loaded sections of all enabled FLASH devices:

```
Memtool.ProgramSections();
```

Verify programmed sections and check for differences:

```
var Diffs = Memtool.VerifySections("");
```

Finally disconnect target and cleanup:

```
Memtool.DisconnectTarget();  
Memtool.Cleanup();
```

Application Notes

JScript Example

```
//
// UDEMemtool automation basics - JScript demo script
//
// On command line run: cscript MemtoolAutomationDemo.js
//

WScript.echo("UDEMemtool JScript Sample");

var ProgId = "UDE.Memtool";

// get current directory
var fso = new ActiveXObject("Scripting.FileSystemObject");
var WorkDir = fso.GetAbsolutePathName(".");

var CfgFile = WorkDir + "\\TriBoard_TC27x_dap.cfg";
var HexFile = WorkDir + "\\MulticoreDemo.hex";
var LogFile = WorkDir + "\\Memtool.log";

WScript.Echo("Create new UDEMemtool object ...");
var Memtool = new ActiveXObject(ProgId);
WScript.echo(" Version: " + Memtool.Version);

WScript.echo("Prepare logging...");
Memtool.CreateLogFile(LogFile, 50);

WScript.echo("Init memtool...");
Memtool.Init(CfgFile);

WScript.echo("Connect target...");
Memtool.ConnectTarget();

WScript.echo("Load target application file...");
Memtool.OpenFile(HexFile, true);

WScript.echo("Program FLASH...");
Memtool.ProgramSections();

WScript.echo("Verify FLASH...");
var Diffs = Memtool.VerifySections("");
WScript.echo(" " + Diffs + " differences found");

WScript.echo("Disconnect and cleanup...");
Memtool.DisconnectTarget();
Memtool.Cleanup();

WScript.echo("finished");
```

Application Notes

VBScript Example

```
'  
' UDEMemtool automation basics - VBScript demo script  
'  
' On command line run: cscript MemtoolAutomationDemo.vbs  
'
```

```
WScript.echo "UDEMemtool VBScript Sample"
```

```
ProgId = "UDE.Memtool"
```

```
dim fso  
set fso = CreateObject("Scripting.FileSystemObject")  
WorkDir = fso.GetAbsolutePathName(".")
```

```
CfgFile = WorkDir & "\TriBoard_TC27x_dap.cfg"  
HexFile = WorkDir & "\MulticoreDemo.hex"  
LogFile = WorkDir & "\Memtool.log"
```

```
WScript.Echo "Create new UDEMemtool object ..."
```

```
dim Memtool  
set Memtool = CreateObject(ProgId)  
WScript.echo " Version: " & Memtool.Version
```

```
WScript.echo "Prepare logging..."  
Memtool.CreateLogFile LogFile, 50
```

```
WScript.echo "Init memtool..."  
Memtool.Init(CfgFile)
```

```
WScript.echo "Connect target..."  
Memtool.ConnectTarget()
```

```
WScript.echo "Load target application file..."  
Memtool.OpenFile HexFile, true
```

```
WScript.echo "Program FLASH..."  
Memtool.ProgramSections()
```

```
WScript.echo "Verify FLASH..."  
Diffs = Memtool.VerifySections("")  
WScript.echo " " & Diffs & " differences found"
```

```
WScript.echo "Disconnect and cleanup..."  
Memtool.DisconnectTarget()  
Memtool.Cleanup()
```

```
WScript.echo "finished"
```

Application Notes

Perl Example

```
#
# UDEMemtool automation basics - Perl demo script
#
# On command line run: perl MemtoolAutomationDemo.pl
#

#!/usr/bin/perl

use strict;
use Win32::OLE;
use Cwd;

print("UDEMemtool Perl Sample\n");

my $ProgId = "UDE.Memtool";

my $WorkDir = cwd();

my $CfgFile = $WorkDir . "\\TriBoard_TC27x_dap.cfg";
my $HexFile = $WorkDir . "\\MulticoreDemo.hex";
my $LogFile = $WorkDir . "\\Memtool.log";

print("Create new UDEMemtool object ...\n");
my $Memtool = new Win32::OLE($ProgId) or die "Failed to create UDE.Memtool object !\n";
print(" Version: " . $Memtool->Version . "\n");

print("Prepare logging...\n");
$Memtool->CreateLogFile($LogFile,50);

print("Init memtool...\n");
$Memtool->Init($CfgFile);

print("Connect target...\n");
$Memtool->ConnectTarget();

print("Load target application file...\n");
$Memtool->OpenFile($HexFile, 1);

print("Program FLASH...\n");
$Memtool->ProgramSections();

print("Verify FLASH...\n");
my $Diffs = $Memtool->VerifySections("");
print(" " . $Diffs . " differences found\n");

print("Disconnect and cleanup...\n");
$Memtool->DisconnectTarget();
$Memtool->Cleanup();

print("finished\n");
```

Application Notes

Python Example

```
#
# UDEMemtool automation basics - Python demo script
#
# On command line run: python MemtoolAutomationDemo.py
#

import sys
import os
import win32com.client

print("UDEMemtool Python Sample")

ProgId = "UDE.Memtool"

WorkDir = os.path.abspath(".")

CfgFile = WorkDir + "\\TriBoard_TC27x_dap.cfg"
HexFile = WorkDir + "\\MulticoreDemo.hex"
LogFile = WorkDir + "\\Memtool.log"

print("Create new UDEMemtool object ...")
Memtool = win32com.client.Dispatch(ProgId)
print(" Version: " + Memtool.Version)

print("Prepare logging...")
Memtool.CreateLogFile(LogFile,50)

print("Init memtool...")
Memtool.Init(CfgFile)

print("Connect target...")
Memtool.ConnectTarget()

print("Load target application file...")
Memtool.OpenFile(HexFile, True)

print("Program FLASH...")
Memtool.ProgramSections()

print("Verify FLASH...")
Diffs = Memtool.VerifySections("")
print(" " + str(Diffs) + " differences found")

print("Disconnect and cleanup...")
Memtool.DisconnectTarget()
Memtool.Cleanup()

print("finished")
```

Application Notes

C# Example

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

// ToDo: Add UDENetCOMLib.UDELib to references
using UDELib;

namespace MemtoolAutomationDemo
{
    class MemtoolAutomationDemo
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("UDEMemtool C# Sample");

            string WorkDir = Environment.CurrentDirectory;

            string CfgFile = WorkDir + @"\TriBoard_TC27x_dap.cfg";
            string HexFile = WorkDir + @"\MulticoreDemo.hex";
            string LogFile = WorkDir + @"\Memtool.log";

            System.Console.WriteLine("Create new UDEMemtool object ...");
            UDELib.UDEMemtool Memtool = new UDELib.UDEMemtool();
            System.Console.WriteLine("  Version: {0}", Memtool.Version);

            System.Console.WriteLine("Prepare logging...");
            Memtool.CreateLogFile(LogFile, 50);

            System.Console.WriteLine("Init memtool...");
            Memtool.Init(CfgFile);

            System.Console.WriteLine("Connect target...");
            Memtool.ConnectTarget();

            System.Console.WriteLine("Load target application file...");
            Memtool.OpenFile(HexFile, 1);

            System.Console.WriteLine("Program FLASH...");
            Memtool.ProgramSections();

            System.Console.WriteLine("Verify FLASH...");
            int Diffs = Memtool.VerifySections("");
            System.Console.WriteLine("  {0} differences found", Diffs);

            System.Console.WriteLine("Disconnect and cleanup...");
            Memtool.DisconnectTarget();
            Memtool.Cleanup();

            System.Console.WriteLine("finished");
        }
    }
}
```