

Als vervolg op het eerste programmaatje om via C++ de VMware Server omgeving te bestieren, dit als gevolg op een wens om Hot-Backups te maken van m'n VMware machines, ben ik verder gegaan met de aansturing van een VMware Server 2.0 d.m.v. de VixApi want deze is wezelijk anders dan die voor VMware Server 2.0. In mijn huidige omgeving heb ik beide smaken draaien, vandaar dit gevolg. Als eerste ben ik begonnen om met Visual C++2008 een programma te vinden waarmee ik de VixApi toegang remote kon test. Uiteindelijk wil ik ook een ontwikkelmogelijkheid/uitvoeringsmogelijkheid op de Ubuntu server hebben maar met de Linux console ben ik nog niet zo vertrouwd. Dus eerst de ontwikkeling op Windows. Als uitgangspunt heb ik het poweron.c script uit de samples directory gebruikt. Deze een beetje gestrip zodat het duidelijk voor me wordt. Verder is het script ook wat verder doorgevoerd, niet alleen een connectietest maar een volledige startup procedure van een server. Bij dit script staan de inloggegevens in het script, niet zo netjes, dat bouw ik de komende dagen nog wel om. Een soort script/programmeer frameworkje moet hier uitkomen. (helaas is mijn vakantie daarvoor te kort)

Hier de source code:

```
#include <stdio.h>
#include <stdlib.h>
#include "vix.h"
#define CONNTYPE VIX_SERVICEPROVIDER_VMWARE_VI_SERVER
#define HOSTNAME "https://ubuntuhost:8333/sdk"
#define HOSTPORT 0
#define USERNAME "XXXXXXXXXXXXXX"
#define PASSWORD "XXXXXXXXXXXXXX"
#define VMPOWEROPTIONS VIX_VMPOWEROP_NORMAL
#define VMXPATH_INFO "where vmxpath is an absolute path to the .vmx file "
"for the virtual machine."

static char *progName;
////////////////////////////////////
static void
usage()
{
    fprintf(stderr, "Usage: %s <vmxpath>n", progName);
    fprintf(stderr, "%sn", VMXPATH_INFO);
}

////////////////////////////////////
int
main(int argc, char **argv)
{
```

```
    VixError err;
    char *vmxPath;
    VixHandle hostHandle = VIX_INVALID_HANDLE;
    VixHandle jobHandle = VIX_INVALID_HANDLE;
    VixHandle vmHandle = VIX_INVALID_HANDLE;

    progName = argv[0];
    if (argc > 1) {
        vmxPath = argv[1];
    } else {
        usage();
        exit(EXIT_FAILURE);
    }

    jobHandle = VixHost_Connect(VIX_API_VERSION,
        CONNTYPE,
        HOSTNAME,*hostName,
        HOSTPORT,*hostPort,
        USERNAME,*userName,
        PASSWORD,*password,
        0,options,
        VIX_INVALID_HANDLE,*propertyListHandle,
        NULL,*callbackProc,
        NULL,*clientData);
    err = VixJob_Wait(jobHandle,
        VIX_PROPERTY_JOB_RESULT_HANDLE,
        &hostHandle,
        VIX_PROPERTY_NONE);
    if (VIX_FAILED(err)) {
        printf("Connectie Server is mislukt!\n");
        goto abort;
    }
    printf("Connectie Server is gelukt!\n");

    Vix_ReleaseHandle(jobHandle);

    jobHandle = VixVM_Open(hostHandle,
        vmxPath,
        NULL,VixEventProc *callbackProc,
        NULL);void *clientData);
    err = VixJob_Wait(jobHandle,
        VIX_PROPERTY_JOB_RESULT_HANDLE,
        &vmHandle,
        VIX_PROPERTY_NONE);
    if (VIX_FAILED(err)) {
        printf("Connectie VmClient is mislukt!\n");
```

```

        goto abort;
    }
    printf("Connectie VmClient is gelukt!\n");
    Vix_ReleaseHandle(jobHandle);
    jobHandle = VixVM_PowerOn(vmHandle,
        VMPOWEROPTIONS,
        VIX_INVALID_HANDLE,
        NULL, *callbackProc,
        NULL, *clientData);
    err = VixJob_Wait(jobHandle, VIX_PROPERTY_NONE);
    if (VIX_FAILED(err)) {
        printf("Connectie VmClient-Poweron is mislukt!\n");
        goto abort;
    }
    printf("Connectie VmClient-Poweron is gelukt!\n");

```

abort:

```

    Vix_ReleaseHandle(jobHandle);
    Vix_ReleaseHandle(vmHandle);
    VixHost_Disconnect(hostHandle);

```

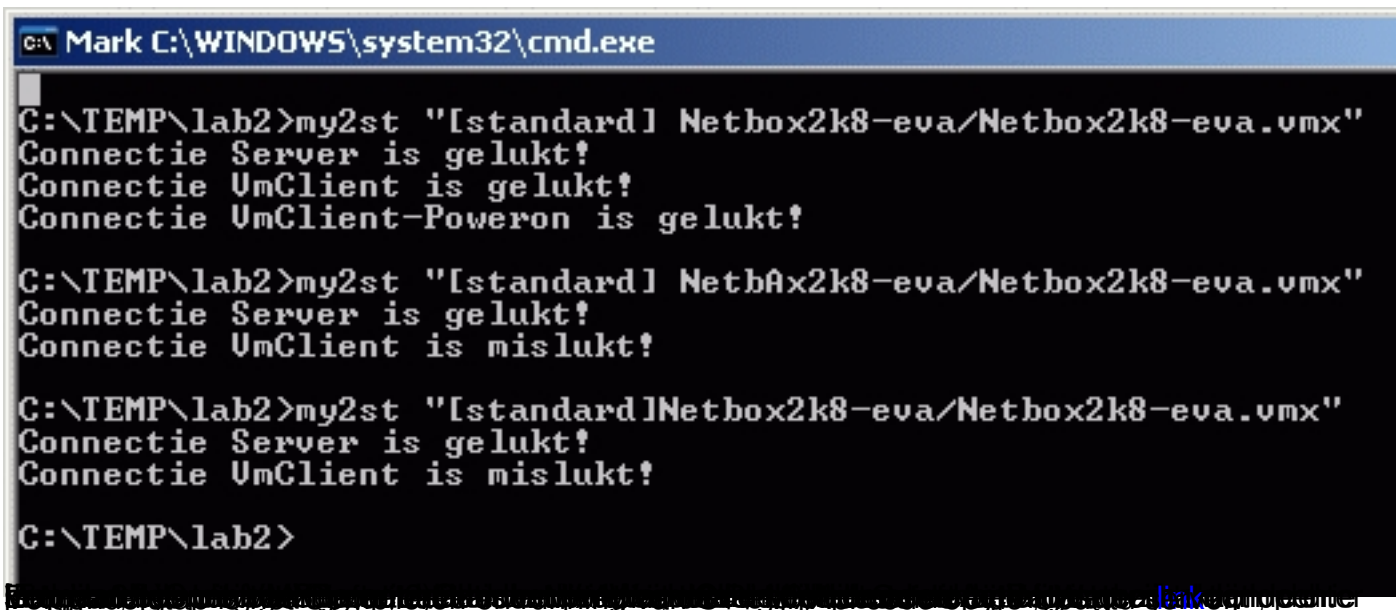
```

    return 0;
}

```

Wees niet bevreesd, het gaat hier geen gewoonte worden om lappen met code te plaatsen, de twee stukken c++ code die hier nu zijn te vinden bevatten de basis om te connecten naar een vmware server via VixApi, in het vervolg zal de daarvoor benodigde code worden weggelaten en alleen flarden relevante code worden geplaatst. Leesbaarheid hou ik hoog in het vaandel.

Wat is er mogelijk met bovenstaande code? Zie daarvoor schermafdruk hieronder. (ps. programma heet my2st)



```
C:\ Mark C:\WINDOWS\system32\cmd.exe

C:\TEMP\lab2>my2st "[standard] Netbox2k8-eva/Netbox2k8-eva.vmx"
Connectie Server is gelukt!
Connectie UmClient is gelukt!
Connectie UmClient-Poweron is gelukt!

C:\TEMP\lab2>my2st "[standard] NetbAx2k8-eva/Netbox2k8-eva.vmx"
Connectie Server is gelukt!
Connectie UmClient is mislukt!

C:\TEMP\lab2>my2st "[standard] Netbox2k8-eva/Netbox2k8-eva.vmx"
Connectie Server is gelukt!
Connectie UmClient is mislukt!

C:\TEMP\lab2>
```