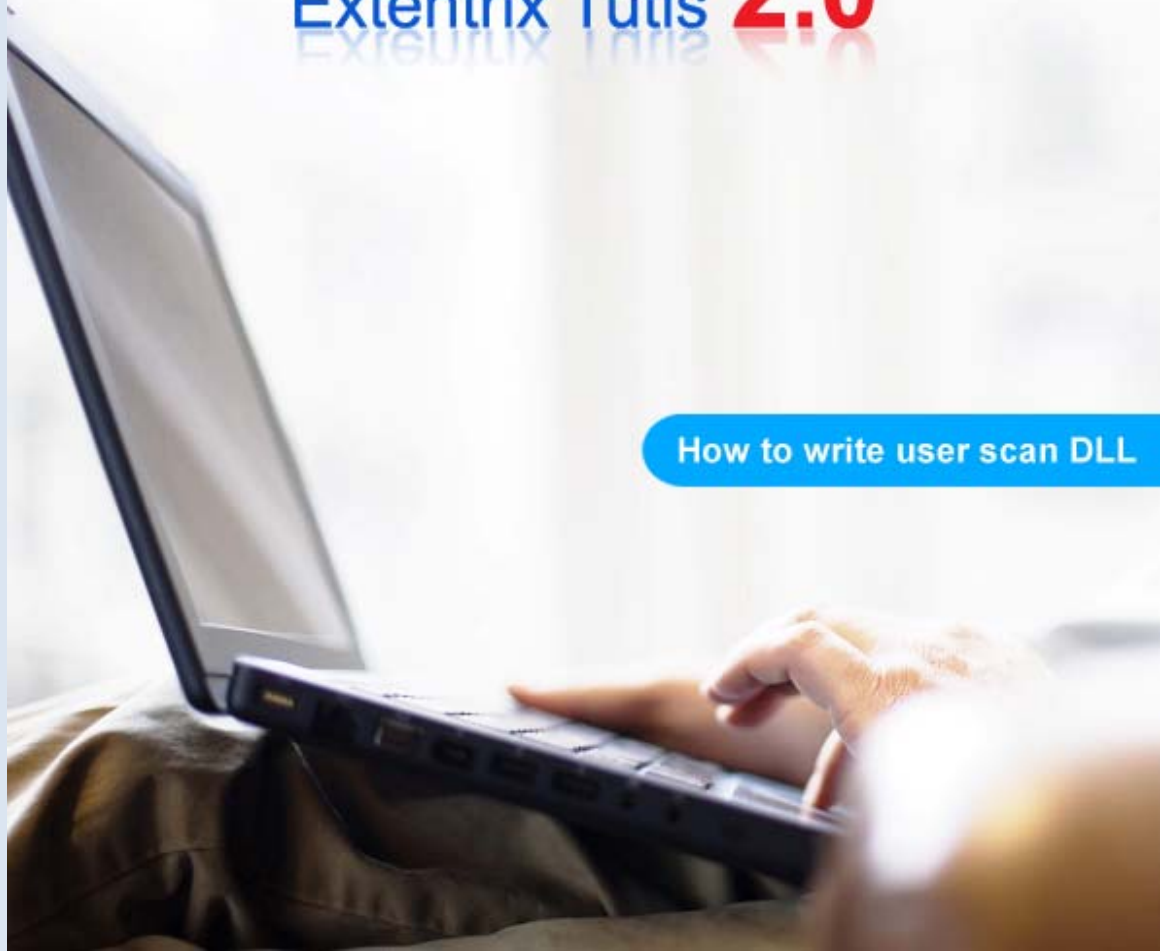


extentrix



# Extentrix Tutis **2.0**

How to write user scan DLL



## NOTICE

Extentrix Systems, FZE, makes no representations or warranties with respect to the contents or use of this publication. Extentrix specifically disclaims any express or implied warranties, merchantability, or fitness for any particular purpose. Extentrix reserves the right to make any changes in specifications and other information contained in this publication without prior notice and without obligation to notify any person or entity of such revisions or changes.

© Copyright 2008-2009 All Rights Reserved

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information and retrieval systems for any purpose other than the purchase's personal use without express written permission of:

Extentrix Systems, FZE

P.O.Box 4404

Dubai, UAE

<http://www.extentrix.com>

**CONTENTS**

How to write user scan DLL ..... 3

- Extentrix scan implementation rules..... 3
- Steps To write Custom User Scan ..... 3
- Custom User Scan SAMPLE..... 4
  - Extentrix Computer Map Check Scan ..... 4

How To Write XML file..... 8

- Extentrix Tag..... 8
- Sample XML File ..... 10
  - Extentrix Computer Map Check Scan XML File..... 10
  - Extentrix Connection Type Scan XML File..... 10

About Extentrix..... 11

## HOW TO WRITE USER SCAN DLL

Extentrix Systems empowers administrators with the ability to develop their own scans. If you have a special scan requirement that Extentrix Tutis does not support by default, you can develop your own custom DLL and prepare the required XML files. Extentrix Management Console allows you to import your custom scans which become part of the Management console.

### EXTENTRIX SCAN IMPLEMENTATION RULES

- The user scan can include more than one function; all should be exported on the same user scan DLL.
- All functions should return a Boolean value (TRUE or FALSE) or equivalent.
- Each function can take a maximum of ten parameters.
- Function parameters can be one of the following types: String, Integer, List (null delimited string).
- Strings and Lists are sent as wide character strings.
- Functions are called using (cdecl) calling convention.

### STEPS TO WRITE CUSTOM USER SCAN

- 1- Start C++ project, and create a Dynamic Library (.dll) project.
- 2- Implement your functionality as C++ code, during implementation take care Extentrix scan implementation rules, maximum you can accept ten parameters for each function and type of this parameter can be one of three options (Integer, String, and List 'null delimiter string').
- 3- Export main function, and make sure that the output of this function is Boolean. To Export the function add the following statement before function definition:

```
extern "C" _declspec(dllexport) bool _cdecl FunctionName(list
parameters)
```

**bool:** mean return type of function is boolean.

- 4- Add resource to solution and fill version information, the import wizard in Tutis Management Console requires the DLL to have a version resource from which the library version is read. To add resource write click on Project, Add, Resource then fill information.
- 5- Build the project, the output must be dll type, and right click on dll file to check version information.

*Extentrix recommended to build the project in Release mode, avoiding failed call depends on client environment.*

*You can develop your own scan using any other language than C++, The only things you should care about that the functions are exported and you have a version resource in your DLL file.*

## CUSTOM USER SCAN SAMPLE

### EXTENTRIX COMPUTER MAP CHECK SCAN

**Scan Name:** ComputerMapCheckScan

**Scan Description:** Scans the user's machine, gets its name, and maps the computer name to one of the allowed groups –if any – where as the scan output reflects the mapping result. If the result shows that the computer map belongs to group specified by the IT administrator, function returns true. Otherwise, returns false.

**Scan Version:** 1.0.0.0

**Scan Parameters:**

Computers Name List: **List Type** each one of its row has a computer name and group pair, separated by (;).

Group Name: **String Type** represents the name of the expected group that the client should belong to.

**Scan Output:**

true – indicates that the client belongs to the specified group.

false – indicates that the client does not belong to the specified group.

**Scan Implantation:**

- 1- Create new C++ project, named ComputerMapCheckScan, Output type is Dynamic Library (.dll).
- 2- Add Resource to ComputerMapCheckScan, fill version information (1.0.0.0).
- 3- Implementation Code:

```
// To export function add the following statement
//function name: CheckComputerMap
```

```

//return type: bool
//Parameters : ComputerMap(null delimiter string) ,
// desired Group name (String)

extern "C"_declspec(dllexport) bool _cdecl CheckComputerMap(wchar_t *
pwcharComputerMap, wchar_t * pwcharGroupName)
{
    bool matched = false;
    bool inGroup = false;
    char sms [1024];

// Get the name of the computer.
// loading kernel32.dll , get the GetComputerNameW method address , and call it

    HMODULE lib = LoadLibrary("KERNEL32.DLL");
    if(!lib)
    {
        // failed to load Library return false
        return false;
    }

    // get method address
    GCN proc = (GCN) GetProcAddress(lib, "GetComputerNameW");
    LPWSTR lpszSystemInfo; //pointer to system information string
    DWORD nsize = BUFSIZE; //size of computer name
    //in case of valid address apply function call
    if(proc)
    {
        // get the computer name if the call was successful and assign it to
        // result,then free the library
        lpszSystemInfo = new OLECHAR[BUFSIZE];
        if(proc(lpszSystemInfo,&nsize))
        {
            // you can view computer name.
        }
    }

    else
    {
        FreeLibrary(lib);
        return false;
    }

    wchar_t * p;
    int i,size,nTokenCount;

    //Enumerate all Computer Name in the Map
    //As we listed Computer name is null delimiter string, have 2 column
    //(ComputerName;expected GroupName) separated by (;)
    //
    //Enumerate all Computer Name in the Map
    //
    for ( p = pwcharComputerMap, i = 0; *p != NULL; i++ )
    {
        wchar_t* pNextToken;

```

```

wchar_t* ComputerNameFromMap;
wchar_t ComputerName[MAX_COMPUTER_SIZE];
nTokenCount = 0;
//
//Tokenize the ComputerMap ComputerName;GroupName
//
size = (int) wcslen(p);
ComputerNameFromMap = wcstok_s( p, L";", &pNextToken );

wcsncpy_s(ComputerName, MAX_COMPUTER_SIZE,
ComputerNameFromMap, MAX_COMPUTER_SIZE);

while ( ComputerNameFromMap && nTokenCount!= 1 )
{
    ComputerNameFromMap = wcstok_s( NULL, L";",
    &pNextToken );
    nTokenCount++;
}

if( ComputerNameFromMap == NULL)
{
    p += size + 1;
    continue;
}

matched = ( 0 ==_wcsicmp(lpszSystemInfo, ComputerName) );

if( !matched )
{
    p += size + 1; //get next row from list
    continue;
}

// set if at least one address matches the group
inGroup = (0 == _wcsicmp(ComputerNameFromMap, pwcharGroupName));

if(inGroup)
{
    delete []lpszSystemInfo;
    FreeLibrary(lib);
    return true;
}

p += size + 1;// move the pointer to next Computer Name
}

delete []lpszSystemInfo;
FreeLibrary(lib);
return false;
}

```

**Scan Build:** Build scan project, in Release mode, checks DLL file properties, it should be have version number.

*If you need to update a custom scan DLL using Extentrix Tutis Management Console, then you should increase DLL version number.*

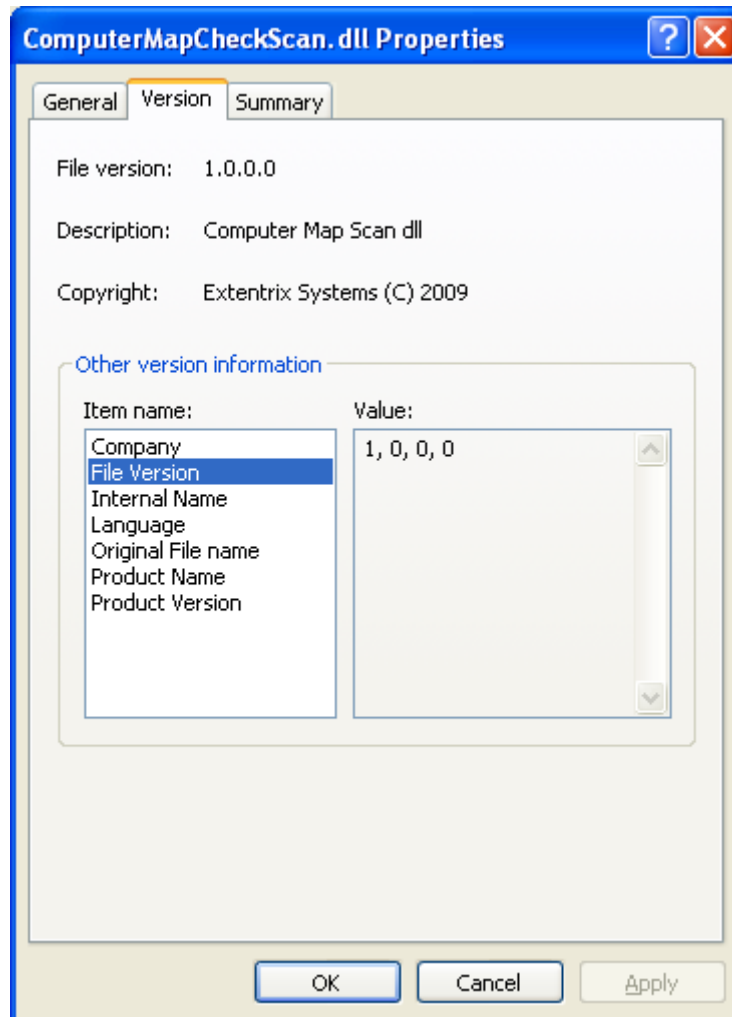


Figure 1: Extentrix Tutis 2.0 Custom Scan (DLL file properties)

After preparing DLL file, we should write XML file.

## HOW TO WRITE XML FILE

You should write an XML description file for each user scan. This enables the import utility to import the user scan and prepare it so that it can be used later in a convenient way. XML file must follow the Extentrix standard:

### EXTENTRIX TAG

- UserScan tag: each XML description file must contain one and only one UserScan tag which describes the user scan, if the file contains more than one tag only the first one is processed and the others are ignored.

```
<UserScan Name="Scan Name" Description="Scan Description ...">
```

...

```
</UserScan>
```

- Function tag: add a function tag inside the UserScan tag for each function in the user scan DLL, you can add as many tags as you want.

```
<Function Name="Function Name" Description="Function Description ...">
```

...

```
</ Function>
```

If the function does not contain any arguments you can simply close the function tag directly as follows:

```
<Function Name="Function Name" Description="Function Description ..." />
```

- Argument tag: add an Argument tag inside the Function tag for each argument in the function; you can add arguments up to 10 arguments.

The following is an example of an XML description for a simple argument

```
<Argument Name="Argument Name" Type="Integer" Description="First Operand"
MaxLength="4" Default="2323" Hidden="yes" />
```

*Default: if the parameter should has default value.*

*Hidden: Take (yes or no), properties to show parameter in UI or not.*

*If Hidden is set to "yes" then the argument must have a default value.*

*MaxLength:*

---

- Integer Type =4, String = 4000, List=4000

The UI for the argument which will be shown when the administrator adds a user scan item to the policy will be generated automatically for the simple arguments.

Extentrix allows you to write a description for the UI of the argument (How the argument will be appeared to the user when he enters its value in the user scan item). Right now Extentrix allows you to use a drop down list for the argument value, If you want to add a UI description for the argument you should add the tag UI in the Argument tag as follows (<select> is used for the dropdown list):

```
<UI >
  <select >
    <option value="1">Value 1</option >
    <option value="2" Selected="yes">Value 2</option >
    <option value="3">Value 3</option >
    <option value="4">Value 4</option >
  </select >
</UI >
```

---

*Note: Value can be (Integer or String).*

If you omitted UI tag, then UI display for any Argument will depend on argument type:

- Integer: Text box.
- List: Text Area.
- String: Text box.

---

**SAMPLE XML FILE**


---

**EXTENTRIX COMPUTER MAP CHECK SCAN XML FILE**

```

<?xml version="1.0" encoding="utf-8" ?>
  <UserScan Name = "Computer Map Check Scan" Description="Scans the users machine, gets its name, and maps the computer name to one of the allowed. If the result shows that the computer map belongs to the group specified by the IT administrator, the user will be granted the access, otherwise his access will be denied.">
    <Function Name = "CheckComputerMap" Description="Scans the users machine, gets its name and maps the computer name to one of the allowed. If the result shows that the computer map belongs to the group specified by the IT administrator, then return true">
      <Argument Name="Computer Map List" Type="List" MaxLength="4000" Description="List of pair separated by (;) (ComputerName;GroupName) ex.(TEST;WORKGROUP)" />
      <Argument Name="Desired Group name" Type="String" MaxLength="256" Description="Desired Group name" />
    </Function>
  </UserScan>

```

---

**EXTENTRIX CONNECTION TYPE SCAN XML FILE**

```

<?xml version="1.0" encoding="utf-8" ?>
<UserScan Name="Connection Type Scan" Description="Determines the connection type and speed of clients machine.">
  <Function Name="CheckConnectionData" Description="Determines the connection type and speed of client's machine">
    <Argument Name="Connection Type" Type="String" MaxLength="40" Description="Connection Type: Defines the desired connection type to be checked">
      <UI>
        <select>
          <option value="Ethernet" selected="yes">Ethernet</option>
          <option value="LAN Wireless">LAN Wireless</option>
          <option value=" Dial Up ">Dial Up</option>
          <option value=" Others ">Others</option>
        </select>
      </UI>
    </Argument>
  </Function>
</UserScan>

```

## ABOUT EXTENTRIX

Founded in 2006, Extentrix Systems is a leading software development company specializing in virtualization and developing custom software solutions for Citrix and Microsoft customers. Established by former software engineers at Citrix, Extentrix have developed access solutions that simplify and speed users' access to Terminal Services and Citrix platforms.

For more information, please visit the company's Web site at [www.extentrix.com](http://www.extentrix.com) or call +971-4-208-8496.