



Tokenized Serial Numbers

Manual for Helper Tools

Table of Contents

General	3
Create Tokenized Serial Numbers	4
SerialTkn.exe - Manual Operation	4
SerialTkn.exe - Automatic Operation	6
SerialTknDll.dll	7
SerialCheck.dll	9

General

The idea behind **Tokenized Serial Numbers** is to provide every one of your customers with a **UNIQUE** serial number which is based on a secret Token provided by you, and a "User Name" and/or "Company Name" provided by the customer.

The **QSetup Composer** includes a built in dialog for creating **Tokenized Serial Numbers**, However if you provide Serial Numbers to a large number of users and/or you want to fully automate the process, we can provide you with additional Helper Tools.

The **Tokenized Serial Numbers Helper Tools** include a stand alone application to manually and automatically create "Tokenized Serial Numbers" and a special purpose DLL that can be easily linked to any CRM application, or WebSite registration form.

When you purchase a license for the **Tokenized Serial Numbers Helper Tools** we will provide you with a ZIP file (320KB) that include the following files:

- SerialTkn.exe - The stand alone application.
 - SerialTknDll.dll - The special purpose DLL.
 - ReadMe.rtf - Help file on how to use the application and the DLL.
 - Dialogs.rtf - Help file to be used by the application.
-
- For more information please write to us at sales@pantaray.com.

Create Tokenized Serial Numbers

Using our **Tokenized Serial Number Helper Tools** you can select from 3 different options to create Tokenized Serial Numbers.

1. Use the **SerialTkn.exe** program to MANUALLY create Tokenized serial numbers.
2. Use the **SerialTkn.exe** program to AUTOMATICALLY create Tokenized serial numbers.
3. Use the **SerialTknDll.dll** DLL to AUTOMATICALLY create Tokenized serial numbers.

SerialTkn.exe - Manual Operation

Run the **SerialTkn.exe** program.

The program will display the following screen:



The screenshot shows the 'Tokenized Serial Numbers' application window (version 11.0.0.0). The interface is divided into several sections:

- Group Level:** A list of radio buttons from 'No Level' to 'Level-8'. 'No Level' is selected.
- Characters:** A group of radio buttons for 'Digits Only', 'Text Only', and 'Text + Digits'. 'Text + Digits' is selected.
- Length:** A numeric input field set to '16' with up/down arrows.
- Options:** Two checkboxes: 'Add Hyphens' (checked) and 'Case Sensitive' (unchecked).
- Input Fields:** Three text boxes labeled 'Token:', 'User Name:', and 'Company Name:'. The 'Token:' field contains a vertical bar cursor.
- Output:** A text box labeled 'Serial Number:' which is currently empty.
- Buttons:** A 'Copy to Clipboard' checkbox (checked) below the 'Serial Number' field, and a vertical sidebar on the right with 'Help', 'Create Serial No.', and 'Close' buttons.

Define the selections you want.

Enter the User Name and Company Name of the customer.

Click [**Create Serial No.**]

In response the program will display the **Serial Number** as seen in the following image:



If **Copy to Clipboard** is checked then the following text will be placed in the clipboard:

User Name: bill

Company Name: microsoft

Serial Number: IZ82-35WK-7457-R922

You can paste this text into an Email message and send it to your customer.

SerialTkn.exe - Automatic Operation

Create and INI file in the following form:

```
[INPUT]
CharType=2
Length=16
GroupLevel=0
AddHyphens=1
CaseSensitive=0
Token=123456
User=bill
Company=microsoft
```

Save the INI file to your disk while giving it a name as you wish.
Run the **SerialTkn.exe** program with the INI file name as a parameter.

Example: SerialTkn C:\TEMP\TKN.INI

The program will run silently, read the file, add the serial number to the file and close itself.
When the program has finished the file will look like this:

```
[INPUT]
CharType=2
Length=16
GroupLevel=0
AddHyphens=1
CaseSensitive=0
Token=123456
User=bill
Company=microsoft
[OUTPUT]
Serial=IZ82-35WK-7457-R922
```

IMPORTANT

You must provide a token and at least UserName or CompanyName. If not the program will issue an error message and write the same error message to the INI file.

Exaple-1:

```
[INPUT]
CharType=2
Length=16
GroupLevel=0
AddHyphens=1
CaseSensitive=0
[OUTPUT]
Error=You MUST Enter a valid Token!
```

Example-2:

```
[INPUT]
CharType=2
Length=16
GroupLevel=0
AddHyphens=1
CaseSensitive=0
Token=123456
[OUTPUT]
Error=You MUST Enter UserName and/or CompanyName!
```

SerialTknDll.dll

The **SerialTknDll.dll** is another way to automate the production Tokenized serial numbers.

Following is a code sample that demonstrates how to call the "SerialTknDll.dll" from within a DELPHI program.

The DLL can also be used with C/C++ program.

Using this technique you can add to your website a registration form that will provide your customers with Tokenized

Serial Numbers online.

DELPHI Sample

```
type
  _GetSerialToken =
    function(CharType, Len, GrpLvl, AddHyphens, CaseSensitive: byte;
      Token, User, Company, Buf: PChar): PChar; cdecl;

procedure TestSerialTknDll;
var Lib: HModule; P1: pointer; P2: PChar;
    CharType, Len, GrpLvl, AddHyphens, CaseSensitive: byte;
    Token, User, Company, Buf: array[0..255] of char;
begin
  Lib:=LoadLibrary(PChar('SerialTknDll.dll'));
  P1:=GetProcAddress(Lib, 'GetSerialToken');
  if (P1<>nil) then
  begin
    CharType:=2; // 0=Digits-Only 1=Text-Only 2=Digits+Text
    Len:=16; // 6..30 - Number of characters in the Serial Number
    GrpLvl:=0; // 0..8 - Read about Group Level in the "Files" page
    help
    AddHyphens:=1; // 0=false 1=true
    CaseSensitive:=0; // 0=false 1=true

    Token:='123456';
    User:='bill';
    Company:='microsoft';

    P2:=_GetSerialToken(P1)(CharType, Len, GrpLvl, AddHyphens,
      CaseSensitive, Token, User, Company, Buf);
    writeln(P2);
  end;
end;
```

C/C++ Sample

```
#include <stdio.h>
#include <windows.h>

typedef LPTSTR (__cdecl *MYPROC) (byte,byte,byte,byte,byte,
                                  LPTSTR,LPTSTR,LPTSTR,LPTSTR);

LPTSTR Test_GetSerialTkn(char * Buf)
{
    HINSTANCE Lib;
    MYPROC ProcAdd;
    LPTSTR Result=NULL;

    byte CharType=2;          // 0=Digits-Only  1=Text-Only  2=Digits+Text
    byte Len=16;              // 6..30 - Number of characters in the Serial
                              // Number
    byte GrpLvl=0;            // 0..8 - Read about Group Level in the "Files"
                              // page help
    byte AddHyphens=1;        // 0=false  1=true
    byte CaseSensitive=0;     // 0=false  1=true

    Lib = LoadLibrary(TEXT("SerialTknDll.dll"));

    if (Lib != NULL)
    {
        ProcAdd = (MYPROC) GetProcAddress(Lib, TEXT("GetSerialToken"));
        if (ProcAdd != NULL)
        {
            Result = (ProcAdd) (CharType, Len, GrpLvl, AddHyphens, CaseSensitive,
                                "123456", "bill", "microsoft", &Buf);
        }
        FreeLibrary(Lib);
    }

    return Result;
}
```


SerialCheck.dll

The **SerialCheck.dll** may be used by your program to check for the serial code every time you run it.

Following is a code sample that demonstrates how to call the "CheckSerial.dll" from within a DELPHI program.

The DLL can also be used with C/C++ program.

DELPHI Sample

```
type
  _CheckSerialToken = function(Token,User,Company,Serial: PChar): integer;
                      cdecl;

function Test_CheckSerialToken: integer;
var P1: pointer; Lib: HModule;
    Token,User,Company,Serial: array[0..255] of char;
begin
  Lib:=LoadLibrary(PChar('SerialCheck.dll'));
  if (Lib<>0) then
  begin
    P1:=GetProcAddress(Lib, 'CheckSerialToken');
    if (P1<>nil) then
    begin
      Token:='123456';
      User:='bill';
      Company:='microsoft';
      Serial:='9882-3554-7457-0920';
      Result:=_CheckSerialToken(P1)(Token,User,Company,Serial);
      writeln(Result);
    end;
    FreeLibrary(Lib);
  end;
end;
```

C/C++ Sample

```
#include <stdio.h>
#include <windows.h>

typedef int (__cdecl *MYPROC) (LPTSTR, LPTSTR, LPTSTR, LPTSTR);

int Test_CheckSerialToken(void)
{
    HINSTANCE Lib;
    MYPROC ProcAdd;
    int Result=0;

    Lib = LoadLibrary(TEXT("SerialCheck.dll"));

    if (Lib != NULL)
    {
        ProcAdd = (MYPROC) GetProcAddress(Lib, TEXT("CheckSerialToken"));
        if (ProcAdd != NULL)
        {
            Result = (ProcAdd) ("123456", "bill", "microsoft",
                                "9882-3554-7457-0920");
        }
        FreeLibrary(Lib);
    }

    return Result;
}
```