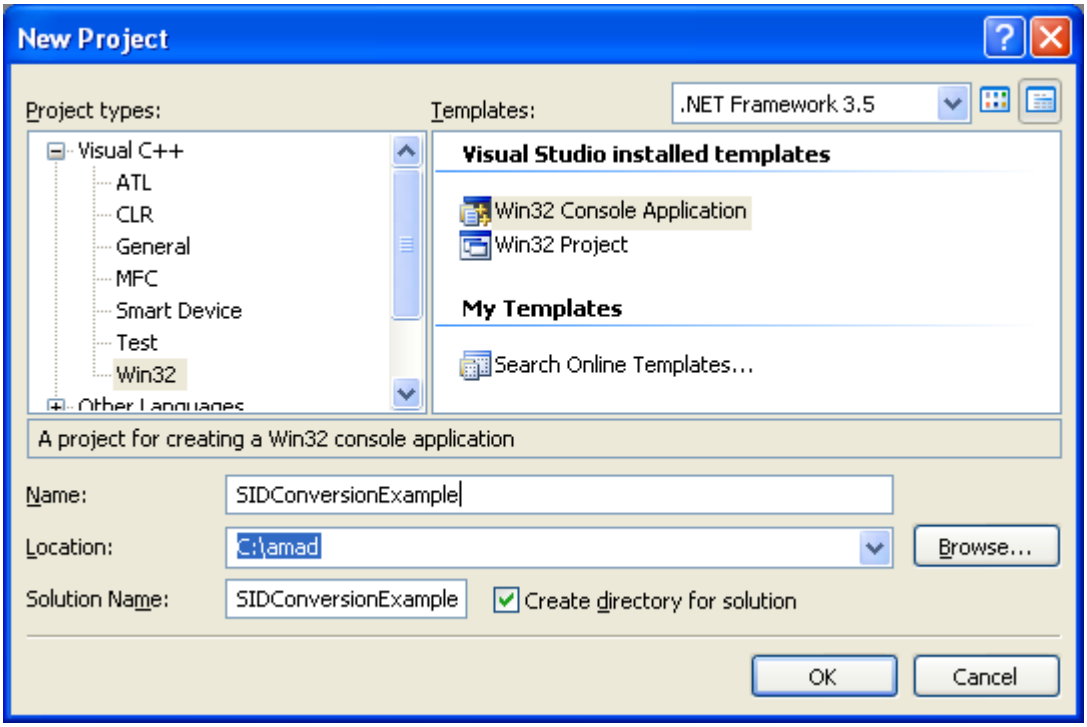
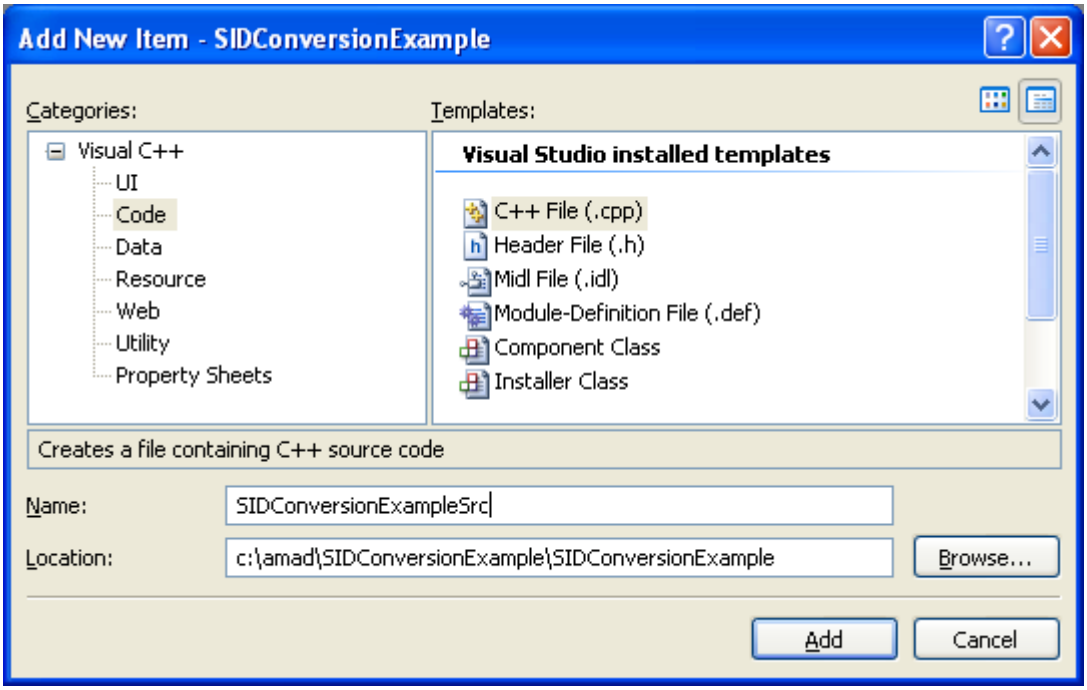
SID conversion: String-to-Binary-to-String Program Example.

The ConvertSidToStringSid() and ConvertStringSidToSid() functions convert a SID to and from string format. For Windows NT 4.0 and earlier the ConvertSidToStringSid() and ConvertStringSidToSid() are not supported.
Create a new empty Win32 console application project. Give a suitable project name and change the project location if needed.



Then, add the source file and give it a suitable name.



Next, add the following source code.

```
// Playing with SID format: Binary SID vs string SID
#include <windows.h>
#include <stdio.h>
#include <sddl.h>
#include <aclapi.h>

int wmain(int argc, WCHAR **argv)
{
	DWORD SidSize, SidSize2;
	PSID TheSID = NULL;
	LPTSTR pSid = L"";

	SidSize = SECURITY_MAX_SID_SIZE;

	printf("Create a well known \"WinLocalSystemSid\" SID.\n");
	printf("-----------------------------------------\n");
	// Allocate ample buffer for the largest possible SID.
	if(!(TheSID = LocalAlloc(LMEM_FIXED, SidSize)))
	{
		wprintf(L"Could not allocate buffer, error %u.\n",
GetLastError());
		// Just exit
		exit(1);
	}
	else
		wprintf(L"Buffer allocated for TheSID successfully.\n");

	// Create a SID for the Local system on the local computer.
	if(!CreateWellKnownSid(
		WinLocalSystemSid,	// Well known Local system SID
```

```
            NULL,                     // Domain SID, NULL for local computer
            TheSID,                         // Pointer to memory for new SID
            &SidSize                  // Pointer in DWORD the number of byte of
TheSid
            ))
      {
            wprintf(L"CreateWellKnownSid() failed, error %u.\n",
GetLastError());
      }
      else
      {
            wprintf(L"CreateWellKnownSid() for Local system is OK.\n");
            wprintf(L"\nConvert the \"WinLocalSystemSid\" SID to string
SID.\n");
            wprintf(L"-------------------------------------------------\n");

            // Get the string version of the SID (S-R-I-I...)
            if(!(ConvertSidToStringSid(
                  TheSID,  // Pointer to the SID structure to be converted
                  &pSid))) // Pointer to variable that receives the null-
terminated SID string
            {
                  wprintf(L"ConvertSidToStringSid() failed, error %u\n",
GetLastError());
                  exit(1);
            }
            else
            {
                  wprintf(L"ConvertSidToStringSid() is OK.\n");
                  wprintf(L"The SID string for WinLocalSystemSid is: %s\n",
pSid);
            }
      }

      if(IsValidSid(TheSID))
            wprintf(L"The SID is valid!\n");
      else
            wprintf(L"The SID is not valid!\n");

      //*********************************************************
      // TODO: Then, use the string SID as needed.
      // ...
      // When done, don't forget to release the buffer used.
      //*********************************************************

      if(LocalFree(TheSID) == NULL)
            wprintf(L"TheSID buffer was freed up...\n");
      else
            wprintf(L"Failed to free up TheSID buffer, error %u\n",
GetLastError());

      //*********************************************************
      LPTSTR StringSid = L"S-1-5-18"; // or "SY" - a well known Local System
      PSID TheSID2 = NULL;
      SidSize2 = SECURITY_MAX_SID_SIZE;

      // S-R-5-18 and equal to...
```

```
	// SECURITY_NT_AUTHORITY\\SECURITY_LOCAL_SYSTEM_RID
	// But they are stored as in binary format in a SID structure
	wprintf(L"\nConvert the \"S-1-5-18\" string SID to SID and then
reconvert.\n");
	wprintf(L"------------------------------------------------------------
\n");
	if(!(TheSID2 = LocalAlloc(LMEM_FIXED, SidSize2)))
	{
		wprintf(L"Could not allocate buffer for TheSID2, error %u.\n",
GetLastError());
		exit(1);
	}
	else
		wprintf(L"Buffer allocated for TheSID2 successfully.\n");

	//*************************************************
	if(!ConvertStringSidToSid(
		StringSid,  // Pointer to a null-terminated string containing the
string-format SID to convert
		&TheSID2))  // Pointer to a variable that receives a pointer to
the converted SID
	{
		wprintf(L"ConvertStringSidToSid() for Local systemfailed, error
%u\n", GetLastError());
		exit(1);
	}
	else
	{
		wprintf(L"ConvertStringSidToSid() for Local system is OK.\n");
	}

	// Re-convert to string SID
	if(!(ConvertSidToStringSid(
		TheSID2,         // Pointer to the SID structure to be converted
		&StringSid)))   // Pointer to variable that receives the null-
terminated SID string
	{
		wprintf(L"ConvertSidToStringSid() again failed, error %u\n",
GetLastError());
		exit(1);
	}
	else
	{
		wprintf(L"ConvertSidToStringSid() is OK.\n");
		wprintf(L"The SID string for WinLocalSystemSid is: %s\n", pSid);
	}

	if(IsWellKnownSid(TheSID2, WinLocalSystemSid))
		wprintf(L"The SID is a well known SID!\n");
	else
		wprintf(L"IsWellKnownSid() failed, error %u.\n", GetLastError());

	//*************************************************
	if(IsValidSid(TheSID2))
		wprintf(L"The SID is valid!\n");
	else
		wprintf(L"IsValidSid() failed, error %u\n", GetLastError());
```

```
	if(LocalFree(TheSID2) == NULL)
		wprintf(L"Buffer for TheSID2 was freed up...\n");
	else
		wprintf(L"Failed to free-up TheSID2 buffer...\n");

	return 0;
}
```

Build and run the project. The following screenshot is a sample output.