

TN1010-1a: CSMail Redistribution Issues for C++ Developers

Contents

1	OLE registration.....	1
1.1	Automatic OLE registration in your install program	1
1.2	Manual OLE registration using regsvr32.exe.....	1
2	Licensing	1

1 OLE registration

Your application uses OLE automation to create an instance of the CSMail client objects and for this to function correctly the CSMail library must be registered with OLE. OLE registration is simply a process of writing some registry values to `HKEY_CLASSES_ROOT` so that OLE can find the library when your program is subsequently executed.

It is important to understand the difference between OLE registration – which makes the library available to OLE and licensing which makes sure that the control is properly licensed on the target machine. OLE Registration is explained in this section and licensing is dealt with in Section 2.

1.1 Automatic OLE registration in your install program

The library is marked as OLE self-registering and most installation programs will perform the OLE registration for you. For example - in InstallShield you should create a File Group for `CSMail.dll` (and any other self registering DLLs that your program uses) and set the `Self-Registered` property of the group to `Yes`. Other installation program will have similar mechanisms for achieving this – try searching for Self-Registering in the help file for your installation program.

1.2 Manual OLE registration using regsvr32.exe

If your installation program is not able to use the self-registration mechanism you can register the control manually using the redistributable `regsvr32.exe` tool provided with your development environment. After copying `csmail.dll` to your target machine you should invoke `regsvr32.exe`:

```
regsvr32 /s "C:\Program Files\My Program\CSMail.dll"
```

This will perform all the steps necessary to make the CSMail library available to OLE automation.

You should find regsvr32.exe in the files installed by your development environment.

2 Licensing

You will have purchased a CSMail license from Codestone Ltd – the library checks the license whenever your program creates an instance of one of the client objects. Your development machine contains a copy of that license so that while you are developing your application you will be able to create instances of the client objects – but what when you need to distribute the control to a computer which does not have a copy of the license?

Visual Basic developers have it easy at this point, the VB compiler will detect that the application is using the CSMail control and automatically compile a copy of the license into the application. C++ developers have to do a bit more work to achieve the same result.

You will probably already have some code in your application to create an instance of, for example, the POP3Client object – it might look something like this:

```
// Create POP3Client object
CSMAILLib::IPOP3ClientPtr ipPop;
HRESULT hR=ipPop.CreateInstance(__uuidof(CSMAILLib::POP3Client));
```

You need to change this code to create an instance of the object using IClassFactory2 (which supports licensing)

```
// Change this to your license code!
#define CSMAIL_LICENSE "abc-1234-1234abcd"
...

CSMAILLib::IPOP3ClientPtr ipPop;
IClassFactory2Ptr ipCF2;

// Get an IClassFactory2 interface
HRESULT hR=CoGetClassObject( __uuidof(CSMAILLib::POP3Client),
                             CLSCTX_ALL,
                             NULL,
                             IID_IClassFactory2,
                             (void **)&ipCF2);

if (SUCCEEDED(hR))
{
    // IClassFactory2 OK - Now create a licensed instance of the object
    hR=ipCF2->CreateInstanceLic( NULL, NULL,
                                __uuidof(CSMAILLib::IPOP3ClientPtr),
                                _bstr_t(CSMAIL_LICENSE),
                                (void **)&ip);
}
}
```

Codestone Ltd
Cheltenham, United Kingdom
Established 1996
Registered in England and Wales
3282460
<http://www.codestone.co.uk/>