

Re: Hosted WinForms Controls and CAS

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2006-02/msg00238.html>

- *From:* Dominick Baier [DevelopMentor] <dbaier@xx>
 - *Date:* Mon, 27 Feb 2006 23:15:49 +0000 (UTC)
-

hi,

this is part of an msi installer project – and should get you started...

```
// this code will run when the MSI file is installed
public override void Install(IDictionary stateSaver) {

// first need to find the machine policy,
// which is where we'll make our changes
PolicyLevel machinePolicy = _findPolicyLevel("Machine");

if (null == machinePolicy) {
// sanity check – this should never happen
throw new ApplicationException("Failed to find the machine policy in the PolicyHierarchy");
}

// we need to add a named permission set
// that includes whatever permissions we're granting
NamedPermissionSet nps = new NamedPermissionSet(permissionSetName, PermissionState.None);
nps.Description = permissionSetDesc;

// TODO: add the permissions AcmeExpense needs
nps.AddPermission(new FileIOPermission(FileIOPermissionAccess.Read, @"c:\acme\expenses"));
nps.AddPermission(new EnvironmentPermission(EnvironmentPermissionAccess.Read,

"EXPENSE"));
nps.AddPermission(new SqlClientPermission(PermissionState.Unrestricted));
nps.AddPermission(new DataProtectionPermission(PermissionState.Unrestricted));

// add our named permission set to the machine policy level
// note that nothing is saved yet (we'll save at the end)
try {
machinePolicy.AddNamedPermissionSet(nps);
}
catch {
// duplicate name – update the existing one with the same name
machinePolicy.ChangeNamedPermissionSet(nps.Name, nps);
}
```

Re: Hosted WinForms Controls and CAS

```
// now we need to create a code group that matches all assemblies
// that we ship with AcmeExpense – one way of doing this is to
// match the strong name we assign to that application (although
// depending on how you manage strong names, this might cover
// a wider set of assemblies)
CodeGroup cg = new UnionCodeGroup(
new StrongNameMembershipCondition(
new StrongNamePublicKeyBlob(acmePublicKey),
null, // match regardless of assembly's simple name
null), // match regardless of assembly's version
new PolicyStatement(nps,
PolicyStatementAttribute.Nothing) // no LevelFinal or Exclusive attribute on this code group
);
cg.Name = codeGroupName;
cg.Description = codeGroupDesc;

// code groups with duplicate names are legal, but messy and confusing,
// so we make sure to first remove any existing code groups with our name
_removeCodeGroupsByName(machinePolicy.RootCodeGroup, cg.Name);

// add our new code group (note w
```