

# Re: Hosted WinForms Controls and CAS

---

*Source:* <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2006-02/msg00240.html>

---

- *From:* Shawn Wildermuth <[swildermuth@xxxxxxxxxx](mailto:swildermuth@xxxxxxxxxx)>
  - *Date:* Mon, 27 Feb 2006 15:49:36 -0800
- 

Hello Dominick Baier [DevelopMentor],

THanks a lot Dominick!

Shawn Wildermuth C# MVP, Author and Speaker  
<http://adoguy.com>

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));
```

## Re: Hosted WinForms Controls and CAS

```
// 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);
}
// 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 we've not saved yet).
machinePolicy.RootCodeGroup.AddChild(cg);

// finally, save all changes atomically.
SecurityManager.SavePolicyLevel(machinePolicy);
}
PolicyLevel _findPolicyLevel(string labelWeWant) {
IEnumerator policyLevelEnumerator = SecurityManager.PolicyHierarchy();
PolicyLevel found = null;
while (policyLevelEnumerator.MoveNext()) {
PolicyLevel lvl = (PolicyLevel)policyLevelEnumerator.Current;
if (labelWeWant == lvl.Label) {
found = lvl;
}
}
return found;
}
void _removeCodeGroupsByName(CodeGroup parent, string childName) {
ArrayList codeGroupsToRemove = new ArrayList();
foreach (CodeGroup existingCodeGroup in parent.Children) {
if (childName == existingCodeGroup.Name) {
```

Re: Hosted WinForms Controls and CAS

```
codeGroupsToRemove.Add(existingCodeGroup);  
}  
}  
foreach (CodeGroup cg in codeGroupsToRemove) {  
parent.RemoveChild(cg);  
}  
}
```

---

Dominick Baier – DevelopMentor  
<http://www.leastprivilege.com>

I am workign with a small intranet app for a customer and we've decided to use Hosted WinForms controls for several really complicated code we have. I can host the controls fine (much easier than I thought actually). But I can't do certain work without elevating the CAS and Zone permissions. THIS isn't a concern for the customer. They are happy to include the app in the "Trusted" zone.

I am at the point where I can detect the framework requirements and the security requirement and forward the user to a page to download an installable package to do the security work of elevating permissions. Before I invent my own thing, I wondered if anyone knew of any examples of how to do with in an Installer? I don't want to elevate more permissions than I really need, so any advice about how to elevate CAS permissions for my particular assembly instead of elevating it for the entire zone would be great. The installer is \*not\* installing the assembly with the controls so that we can download new versions as necessary. That might complicate things those. Any hints or urls would help.

BTW, I have googled and found lots of example of how to do the hosting, but not the security side...so don't bother just sending me links to places that explain the <object ... /> tag syntax.

TIA

Shawn Wildermuth  
C# MVP, Author and Speaker  
<http://adoguy.com>