

## Software Documentation : .Net Generic Active Directory ASP.NET Profile Provider 1.0

---

This page last changed on Jul 01, 2008 by [mrgumbe](#).

### 1. Scope

#### 1.1 Overview

The ASP.NET User Providers make it easy to integrate user profile based logic into any application. Unfortunately, the providers only ship with one implementation of these providers: for SQL Server. The Active Directory ASP.NET User Profile Provider component will provide a Profile Provider that uses .NET's Directory Services library to read and write user profile against Active Directory.

##### 1.1.1 Version

1.0

#### 1.2 Logic Requirements

##### 1.2.1 ProfileProvider Implementation

The component will extend from the System.Web.Profile.ProfileProvider class, and be fully compatible with all ASP.NET profile provider functionalities. Each method call will create its own Active Directory connection, which will be used within the context of that call. The component will leverage the classes of the System.DirectoryServices namespace to handle all communications with the active directory server.

##### 1.2.2 Property Mapping

Having a simple Active Directory question is not enough to successfully load a fully configured user profile. The component must have a mapping between property names of the profile, and the corresponding attributes of an account in active directory. The component must support "complex" profile properties as well.

##### 1.2.3 User Directory

The component also needs to know where to find account objects. It must define a property that represents the "root" directory for all users. The component will look in this root directory for all accounts. The component will have a flag that indicates whether all users are at this top level directory (for speed in loading a single named account), or whether users may reside in nested child directories that must be found using a System.DirectoryServices.DirectorySearcher object.

##### 1.2.4 Configuration Mechanism

The component must be fully configurable from app.config / web.config. This can be done by defining its own ConfigurationSection class that will define which properties go in this section. The connection string itself must be placed under the standard "Connection Strings" section of app.config / web.config.

#### 1.3 Required Algorithms

None

#### 1.4 Example of the Software Usage

A company stores detailed profile information in Active Directory. It uses the Active Directory ASP.NET Profile Provider to leverage this data in the standard .NET way.

#### 1.5 Future Component Direction

None identified

## 2. Interface Requirements

### 2.1.1 Graphical User Interface Requirements

None

### 2.1.2 External Interfaces

The component must extend System.Web.Profile.ProfileProvider

### 2.1.3 Environment Requirements

- Development language: C# 2.0
- Compile target: .NET Framework 2.0

### 2.1.4 Package Structure

TopCoder.Web.Profile.Providers.ActiveDirectory

## 3. Software Requirements

### 3.1 Administration Requirements

#### 3.1.1 What elements of the application need to be configurable?

- Connection String
- Path to user objects
- Property mappings and types
- Whether to use a nested search of the user directory

### 3.2 Technical Constraints

#### 3.2.1 Are there particular frameworks or standards that are required?

- LDAP / Active Directory
- ASP.NET Profile Providers

#### 3.2.2 TopCoder Software Component Dependencies:

None

\*\*Please review the TopCoder Software component catalog for existing components that can be used in the design.

#### 3.2.3 Third Party Component, Library, or Product Dependencies:

None

#### 3.2.4 QA Environment:

- Windows XP
- Windows 2000
- Windows 2003
- Active Directory

### 3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines.

## **3.4 Required Documentation**

### **3.4.1 Design Documentation**

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

### **3.4.2 Help / User Documentation**

- Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.