



## CHAPTER 8

---

# *Utilizing Microsoft UDDI Services in your enterprise*

- 8.1 Installing UDDI Services 193
- 8.2 The UDDI Services Console 197
- 8.3 Configuring and using UDDI Services 204
- 8.4 Summary 214

Universal Description, Discovery, and Integration (UDDI) Services is an industry specification created by the UDDI Specification Technical Committee that allows you to publish, discover, share, and reuse web services. For example, if your company wants to publish a public web service on the Internet, you can register it with a UDDI provider such as <http://uddi.microsoft.com> (note that [uddi.microsoft.com](http://uddi.microsoft.com) is one of *many* UDDI providers). Registering your service places your information in a central location where users can search for web services by keyword or topic. In essence, UDDI is a registry and search engine for web services.

Before Windows Server 2003's UDDI Services came along, in order to utilize UDDI you had to publish your web services on the Internet, which for a lot of organizations was not an option. The UDDI server on the Internet was only good for publishing and discovering web services available publicly on the Internet. However, UDDI Services are designed for your local web services that are running on your intranet, which means you can publish and discover web services that are local to your organization. This feature is extremely beneficial to larger organizations because they don't have to compromise security by publishing web services on the Internet but can still take advantage of UDDI.

This chapter begins by outlining the steps required to install UDDI Services. Then, we discuss and configure the properties of a UDDI Services site. Finally, we look at the interfaces (both web and Visual Studio .NET) and show you how to use them to integrate UDDI Services into an intranet.

## 8.1 **INSTALLING UDDI SERVICES**

By default, UDDI Services is not set up when you install Windows Server 2003. This section will guide you through the process of installing and configuring UDDI Services. But first, let's review the required software and hardware. You'll need one of the following:

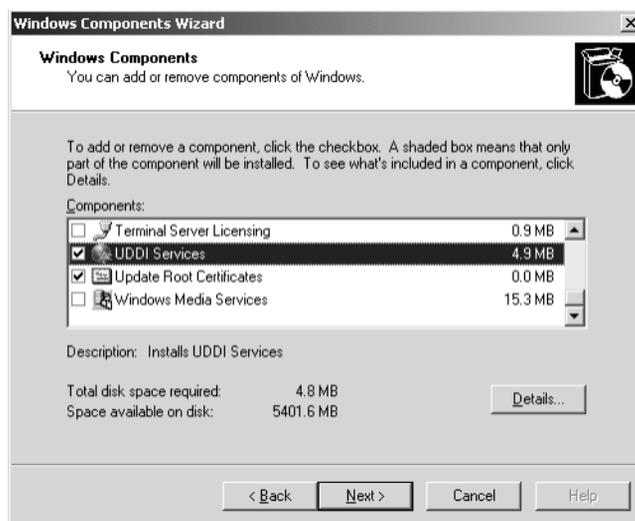
- Windows Server 2003, Standard Edition
- Windows Server 2003, Enterprise Edition
- Windows Server 2003, Datacenter Edition

Notice that UDDI Services is not included as part of Windows Server 2003 Web Edition. The hardware requirements for UDDI Services are:

- 128 MB of RAM (minimum)
- 80 MB of disk space (minimum)

If you meet the software and hardware requirements, you can move on to the next step, which is installing UDDI Services.

As we mentioned earlier, UDDI Services is not installed by default when you set up Windows Server 2003. You can begin the installation process by clicking Add Or Remove Programs in Control Panel and then selecting Add/Remove Windows Components. This opens the Windows Components Wizard, shown in figure 8.1. Here, select all of the components that are associated with UDDI Services and click Next. The



**Figure 8.1**  
You can add or remove Windows components in this dialog box.

UDDI Services component has three subcomponents: Administration Console, Database Components, and Web Server Components. Because this will be the only UDDI Services server in our organization, select all of these options; however, you may want to distribute these options (Administration, Database, and Web Server) over multiple machines for better performance. A distributed installation will work only on Windows Server 2003, Enterprise Edition, and Datacenter Edition.

Next, the wizard asks you to set up a database in which UDDI Services will store its data. This window gives you two options: a Microsoft Data Engine (MSDE) or a SQL Server 2000 database. If you select MSDE (the option that we selected in figure 8.2) and you haven't installed MSDE on your server, your operating system will set up MSDE before UDDI Services and the instance will be named "UDDI".

If you have SQL Server 2000 installed on your server, then you can select a database instance in which to install the UDDI database.

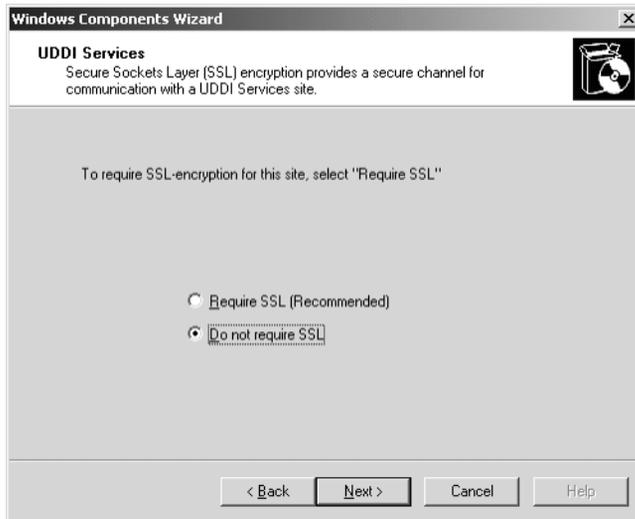
After you make your database selection and click Next, the wizard asks you whether your UDDI web console will use Secure Sockets Layer (SSL) for encryption. If you want to use SSL encryption for your UDDI Services web console, select Require SSL. (Note that the web interface will not work until you configure SSL on your web server.) For our example, choose Do Not Require SSL, as shown in figure 8.3, and click Next.

At this point, you must specify the location(s) in which you want to store your database files. The default is c:\inetpub\uddi\data, as shown in figure 8.4. This means all of the files will be stored in one directory, but you can choose to indicate a location for each database file (database files, log files, backups, etc.) by clicking More. For now, accept the default and click Next.

The next step is to establish security credentials that UDDI Services will use. The default account, as shown in figure 8.5, is Network Service. Network Service is a built-in account that permits limited access to system resources but allows you to communi-



**Figure 8.2**  
Creating a UDDI  
database

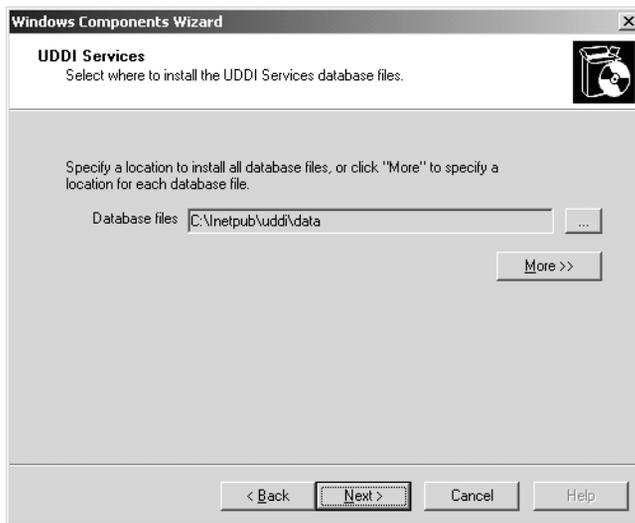


**Figure 8.3**  
You must specify whether you want to use SSL.

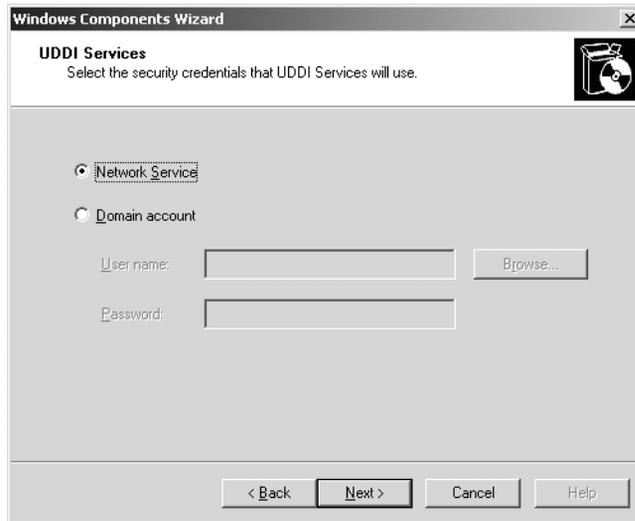
cate over the network. You could specify a domain user account by selecting the second option and completing the User Name and Password fields. For our example, leave the default of Network Service and click Next.

Now, you must choose a name for the UDDI Services site you are creating in this setup process. The name that you specify is the one that you'll use to manage and reference the site with the UDDI Services snap-in application (which you can use to manage multiple UDDI Services sites). For this install, type *EnterpriseUDDI*, as shown in figure 8.6, and click Next.

The next window, shown in figure 8.7, asks you to specify whether you want to automatically register any available site interfaces. You can select this option to automatically



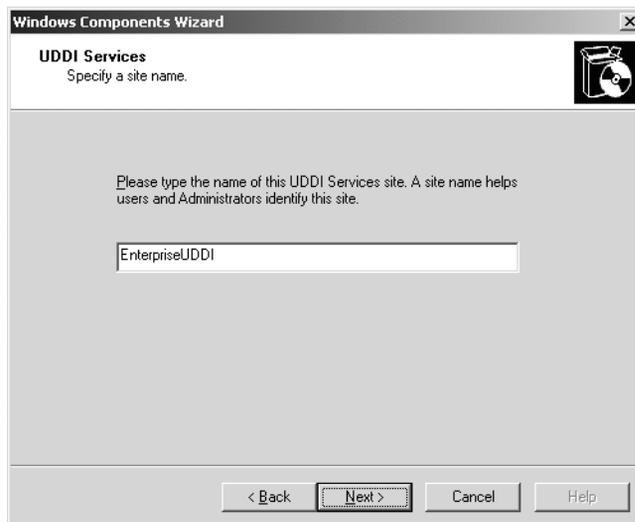
**Figure 8.4**  
Specify a location for the UDDI Services database files.



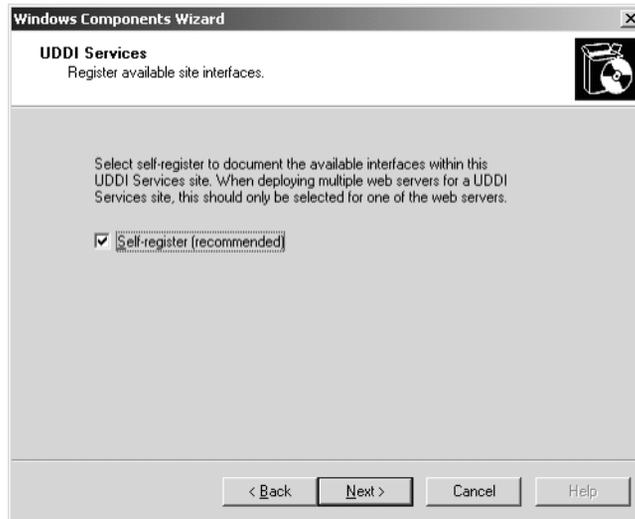
**Figure 8.5**  
UDDI security  
credentials

publish any interfaces and bindings in this site and attempt a silent publication into Active Directory using the current user's credentials. If you are deploying multiple UDDI Services web servers, then this option should be selected for only one of the servers.

As soon as this step has been completed, the UDDI Services installation will begin. When the installation completes, click Finish. UDDI Services should now be set up on your server. Keep in mind that this was a simple, standalone installation. If you would like to perform a distributed installation, please reference the UDDI Services documentation for setup instructions.



**Figure 8.6**  
Specify a UDDI  
Services site  
name.



**Figure 8.7**  
**Registering**  
**available site**  
**interfaces**

## 8.2 THE UDDI SERVICES CONSOLE

When your installation of UDDI Services has completed, you can begin to administer your UDDI Services site using the UDDI Services snap-in tool. As with IIS 6 and COM+, you use the UDDI Services Console to manage the UDDI sites that are running in your enterprise. You can open this tool, shown in figure 8.8, by clicking Start, selecting Administrative Tools, and then clicking UDDI Services.

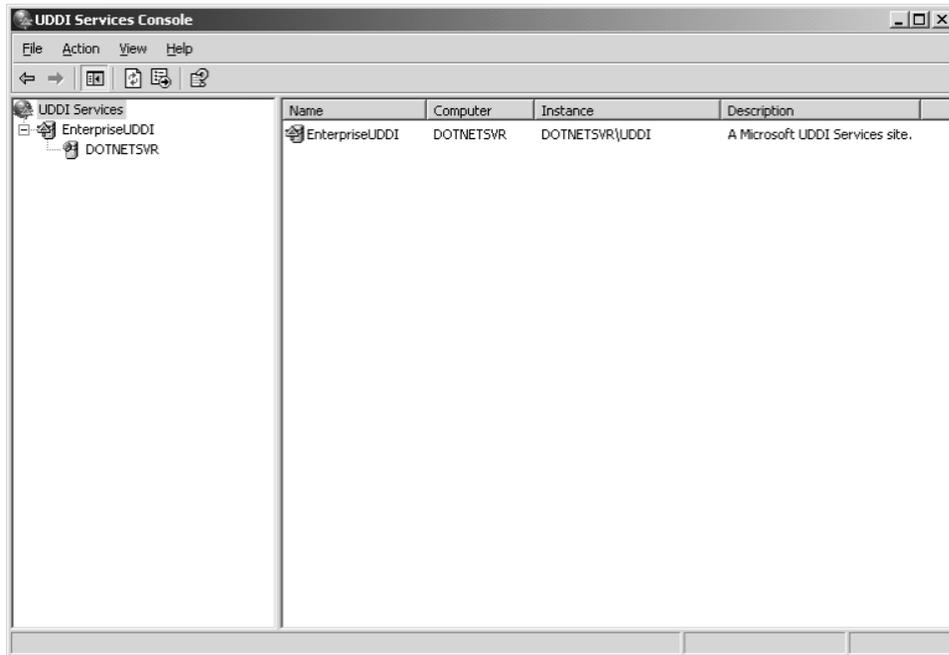
The tool is very simple to navigate because it offers very few options. The first level on the tree view is UDDI Services; this is the root element, and it can contain multiple UDDI sites. Notice the name of the first site, EnterpriseUDDI, which you set up during the UDDI Services installation process. Each site entity can contain multiple web servers. Notice that our web server is called DOTNETSVR.

### 8.2.1 Site properties

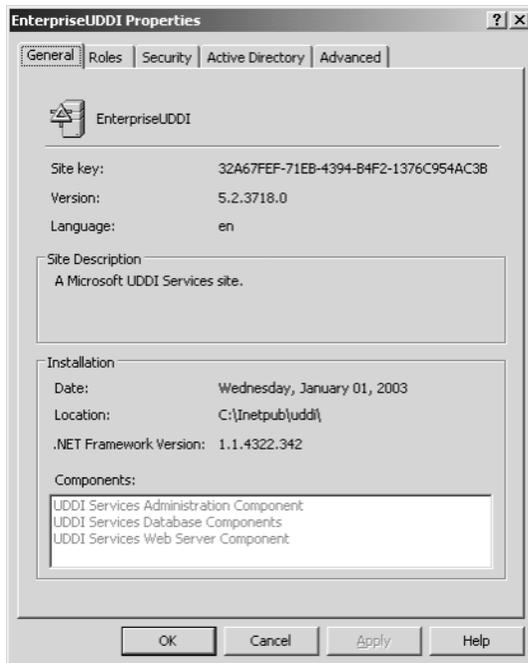
Each entity has multiple properties that you can configure. Let's begin by looking at the properties of the site, EnterpriseUDDI. The site's Properties dialog box has five tabs: General, Roles, Security, Active Directory, and Advanced.

The General tab (figure 8.9) displays general information about the site. The site key is an automatically generated GUID that is assigned to uniquely identify the site. You can use this key for performing programmatic queries against web services contained within your site. Next the tab specifies the version of UDDI Services that is installed on your machine. The Language property identifies the default language code that UDDI Services is configured to use—in this case, en (English).

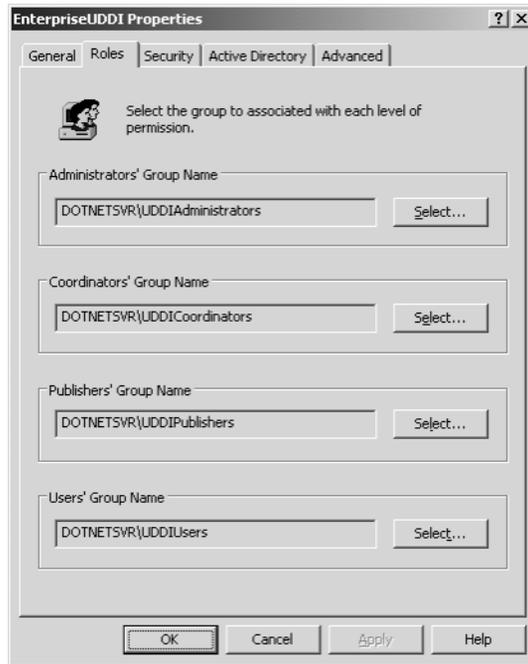
Information about the installation of UDDI Services is also displayed on this tab. This information includes the date of the install, the location of the files used by UDDI Services, and the .NET Framework version that is running on the server.



**Figure 8.8** The UDDI Services Console



**Figure 8.9**  
The General tab  
of the UDDI Site  
Properties box



**Figure 8.10**  
**The Roles tab**

At the bottom of this tab you'll see a list of components that were installed with this site. As you can see in figure 8.9, the Administration, Database, and Web Server components were all successfully installed with this site instance.

The next tab, Roles (figure 8.10), lets you configure various roles for accessing UDDI Services. UDDI Services uses four roles to establish levels of access to the web interface: Administrators, Coordinators, Publishers, and Users. Table 8.1 outlines each role and what actions each can perform.

**Table 8.1 UDDI Services roles**

Activity	Administrators	Coordinators	Publishers	Users
Search for providers, services, and tModels (see section 8.3)	X	X	X	X
Publish providers, services, and tModels	X	X	X	
View server stats and manage entity ownerships	X	X		
Manage categorization schemes	X	X		
Import data	X			
Execute command-line tools	X			
Configure web and database servers, assign groups to roles, configure authentication	X			

In figure 8.10, notice that each role is assigned a specific domain group. This is not the default. The default is to assign the Users role to the BUILTIN\Users domain group and every other role to the BUILTIN\Administrators domain group. For this example, we created new domain groups to accommodate each role. When a user logs on to the web interface and doesn't have access to perform a certain action, all links to that blocked action are hidden from the user. (The web interface will be described in section 8.3.) This is a good practice because it allows you to provide the most flexibility in security.

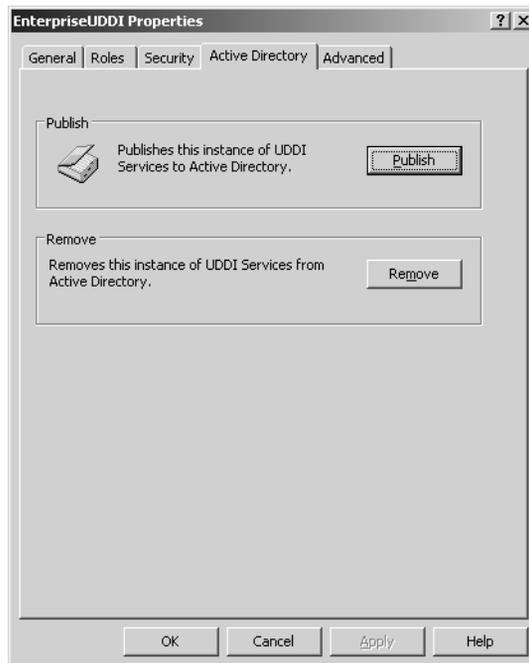
The next tab, Security (figure 8.11), allows you to specify authentication, SSL, and cryptography settings for your site. You can apply one of three types of authentication to your site:

- Windows Integrated And UDDI Publisher Authentication—Authenticate using Windows Integrated *or* UDDI Publisher authentication
- UDDI Publisher Authentication—Authenticate using UDDI Publisher authentication
- Windows Integrated Publisher Authentication—Authenticate using Windows Integrated authentication

You can also enable or disable SSL for your site. Because we chose not to enable it during setup, this checkbox is empty, but you can change that at any time. If you enable SSL, you will need to configure your web server to support certificates.



**Figure 8.11**  
The Security tab



**Figure 8.12**  
**The Active Directory tab**

The last property that you can set on the Security tab is cryptography. If you click the Change button, you will be presented with a dialog box in which you can configure SOAP authentication token expiration and cryptography key timeout.

The next tab in the Properties dialog box is Active Directory, shown in figure 8.12. Once you publish your instance of UDDI Services to Active Directory, users can search your UDDI Services by using Lightweight Directory Access Protocol (LDAP). You can click the Remove button to remove the UDDI Services instance from Active Directory, which will prevent it from being searched.

The final tab, Advanced (figure 8.13), allows you to configure advanced properties for your site. The three advanced properties are:

- Operator—The value used to populate the `Operator` attribute in SOAP API responses
- Find.MaxRowsDefault—The maximum number of rows returned in a search result
- DefaultDiscoveryURL—The default discovery URL that is attached to providers published in UDDI

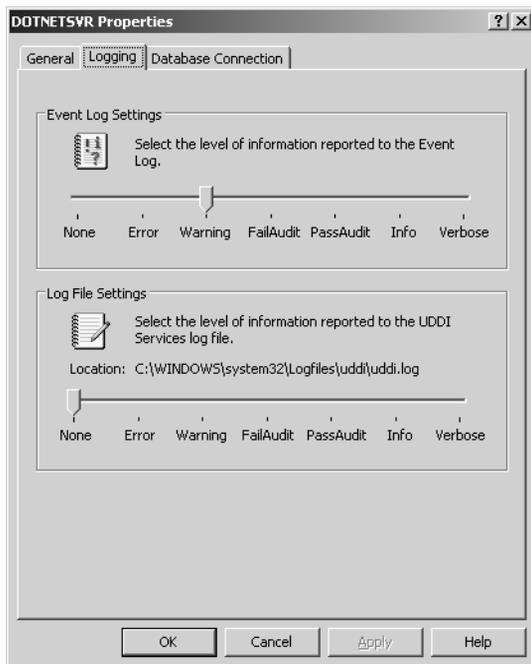
### **8.2.2 Server properties**

Now that we've talked about all of the properties of the site, let's look at the properties of your server. When you install UDDI Services on a server, you can manage the properties of the component(s) that you installed (remember, you don't have to install all

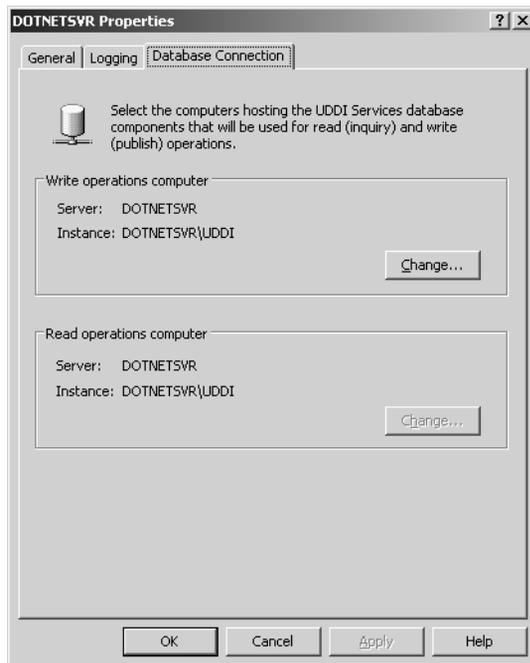




**Figure 8.14**  
The General tab of the UDDI Server Properties dialog box



**Figure 8.15**  
The Logging tab



**Figure 8.16**  
**The Database**  
**Connection tab**

The third tab, Database Connection (figure 8.16), allows you to configure where UDDI Services will store and read its data. You can change these settings to distribute your data to different SQL Servers.

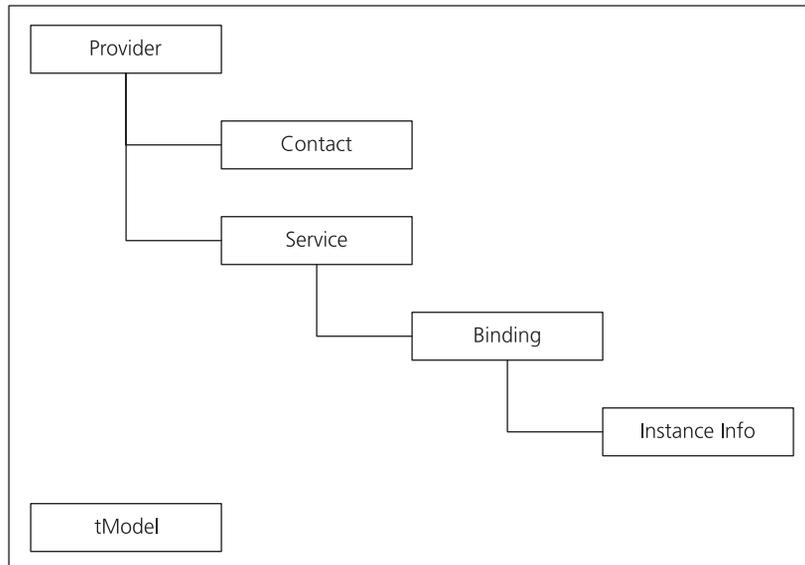
### **8.3 CONFIGURING AND USING UDDI SERVICES**

UDDI Services provides UDDI functionality not only within an enterprise, but also between businesses. Figure 8.17 illustrates how UDDI Services represents organizations and the products and services that they provide.

The base element that you can create in UDDI Services is a Provider element. A *provider* is anyone (business, person, computer, etc.) that provides web services. Providers act as “parent” elements for contacts and services.

The Contact element contains the point of contact for a particular provider. Contacts can provide information about a provider or web services that a provider makes available. Note that a provider can have many contacts associated with it.

The Service element represents the web service that is being exposed in UDDI Services. In our example from the previous chapter, this could represent the `GetAllContacts` method. Under the Service element, the Binding element represents the URL in which your web service (the Service element) is located. Next, the Instance Info element represents a reference to a tModel.



**Figure 8.17 UDDI Services entities**

Finally, the tModel element provides you with access to technical information about your web service. In our examples, this will be the WSDL file that was generated by Visual Studio .NET.

### 8.3.1 A UDDI Services example

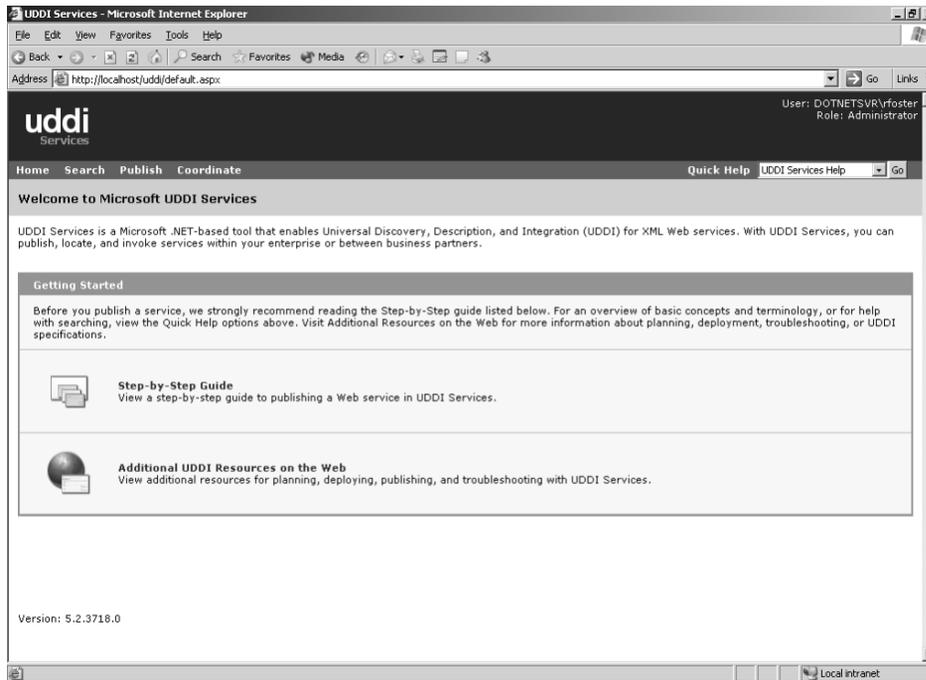
Now that we have discussed the objects and properties of UDDI Services, let's look at an example of utilizing UDDI Services and how you can integrate these services into the web service examples we built in the previous chapter.

Two options that you can use to interact with UDDI Services are a web interface and Visual Studio .NET integration. Let's take a closer look.

#### **The UDDI web interface**

The web interface provides everything that you need to publish, search, and coordinate web services. It consists of two separate sites: one that uses Windows Integrated Security for read-write access and one that uses no security and is for read-only access. The URLs are `http://<serverName>/uddi` (Windows Integrated Security) and `http://<serverName>/uddipublic` (no security). Let's begin by looking at the Windows Integrated Security UDDI web interface. Figure 8.18 shows the UDDI Services web interface. You can get to this screen by opening Internet Explorer and navigating to `http://UDDIservername/uddi/default.aspx`.

Notice that the menu bar near the top of the page offers four options: Home, Search, Publish, and Coordinate. These four options show up because the logged-on user that is viewing this page is in the UDDIAdministrators group (see figure 8.10).



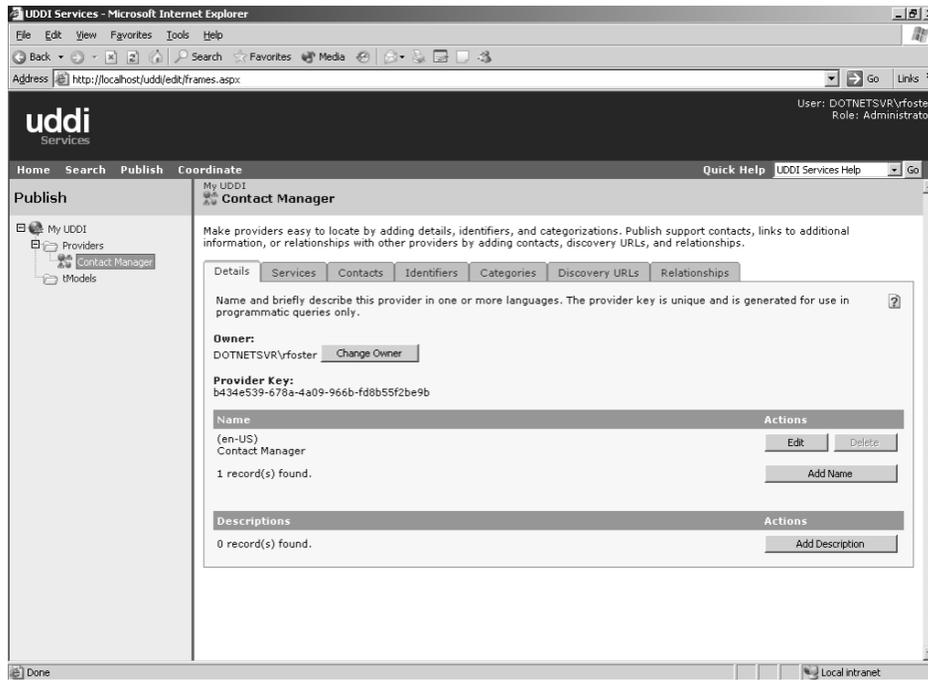
**Figure 8.18** The UDDI web interface

For example, if the logged-on user was in the UDDIPublishers group, they would not have access to coordinate web services and the Coordinate menu item would not be displayed.

You must follow several steps to register a web service in UDDI Services. Before you begin this process, we recommend that you collect as much information as possible about the web services that you plan to register.

The first step is to *publish* a service provider. The service provider is a base element of UDDI Services that contains contacts and services. You begin the process of publishing a provider by clicking the Publish menu item on the UDDI Services web interface (<http://<serverName>/uddi>). This opens the Publish page of UDDI Services. You are presented with three tabs: My UDDI, Providers, and tModels. The My UDDI tab allows you to view the web services, providers, and tModels that you (the currently logged-on user) have published. The Providers tab allows you to manage (add, edit, delete) providers in UDDI Services, and the tModels tab lets you manage tModels for your UDDI Services site. To publish a provider, click the Providers tab, and then click the Add Provider button. This creates a new provider and displays its properties, which you must set manually.

Figure 8.19 displays the screen in which you configure your provider. When a provider is created, a Provider Key (GUID) is generated to uniquely identify your provider



**Figure 8.19** Creating a provider

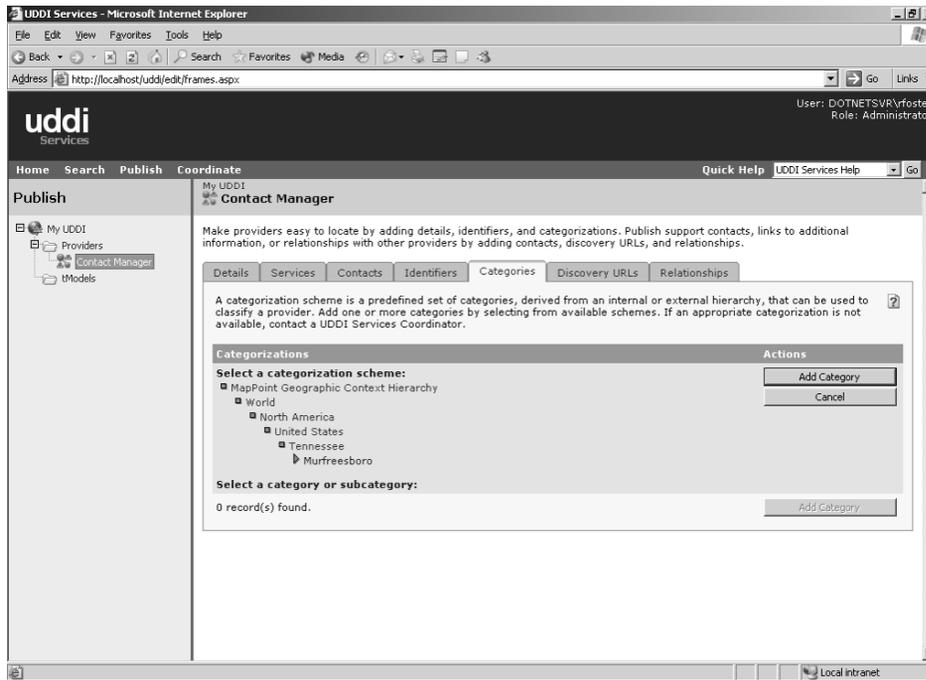
within UDDI Services. Your new provider’s name is initially set to (New Provider Name). You can change this property by clicking the Edit button.

Next (this step is optional), you can add an identifier to your provider by clicking the Identifiers tab. An *identifier* is a reference to a company-wide standard for identifying this provider—a Data Universal Numbering System (DUNS) number, for instance. Identifiers help users when searching and discovering providers within UDDI Services.

The next step is to assign categories to your provider. A *category* can be any set of “categorizations” that you can assign. In figure 8.20, we’re assigning a geographic category to our provider so that it can be searched based on its location. Other categories include UDDI types (postal addresses, namespaces, unique identifiers, etc.), Visual Studio .NET (Encryption, Charting, Communication, etc.), and UDDI relationships (identity, parent-child, peer-to-peer).

Finally, you can (optionally) assign a Discovery URL to your provider that can be used to supply information about the provider. By default, UDDI Services creates one Discovery URL that contains information about your provider in XML format. Additionally, you can create HTML documents that provide “pretty” information about your provider and post the URL(s) to those documents here.

The next step is to create a contact for the provider. The contact will be the person or persons who can be contacted for information about this provider. You can supply



**Figure 8.20** Assigning a category to a provider

information on each contact by clicking the Details tab, shown in figure 8.21. Notice that our contact's name is Chip Andy, and that he is the contact for technical questions regarding our provider. You can also (optionally) specify email addresses, phone numbers, and postal addresses for each contact.

At this point, you can create a Service object in UDDI Services. The Service object represents the XML web service that you wish to publish to UDDI Services. Simply click the Edit button and type a meaningful name for the service that you will be registering. As figure 8.22 shows, we named our new service *ContactsService*. As with the Provider object, you can specify categories in which searches can be performed on your service.

Next, you need to publish a binding for your web service to UDDI Services (figure 8.23). A binding is any access point that you can use to invoke a function on your web service. The number of bindings published represents the number of access points that you wish to provide to your functions and should reflect your company's standards.

When you publish a binding for your web service, provide a URL or pointer to the binding using one of the following protocols:

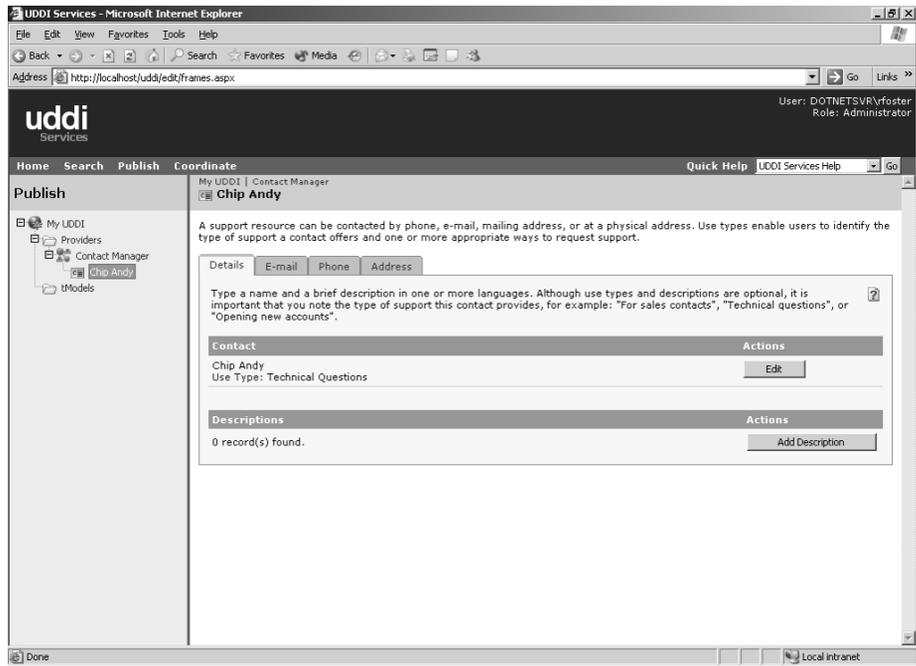


Figure 8.21 Adding a contact

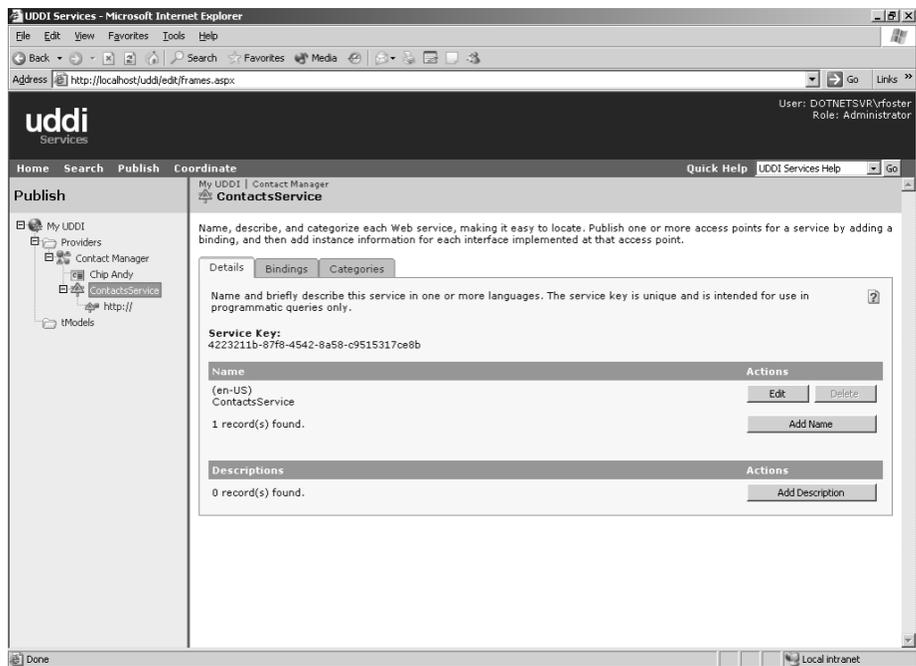


Figure 8.22 Creating a Service object

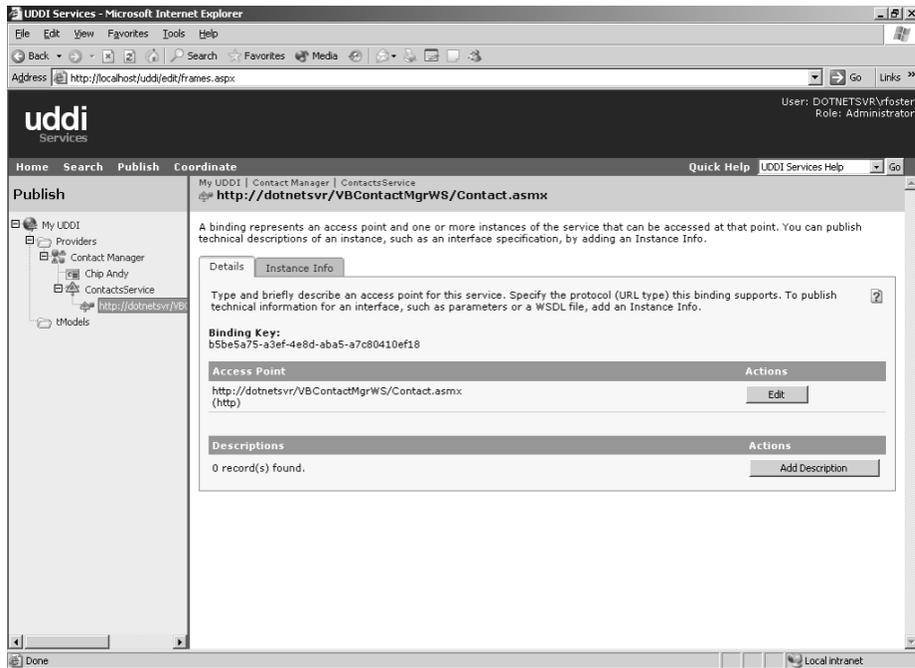
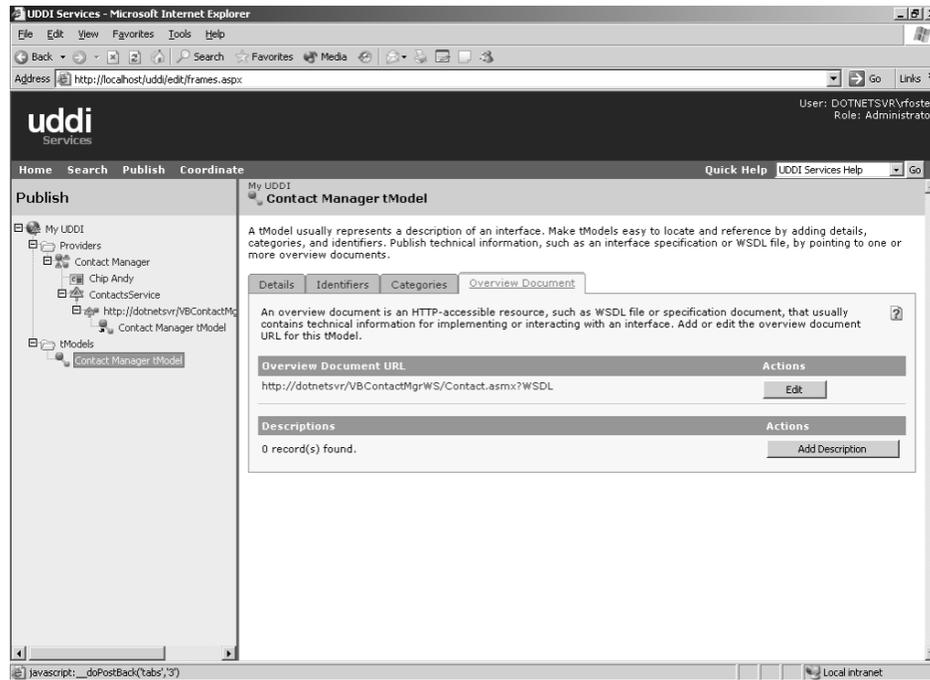


Figure 8.23 Publishing a binding

- Mailto
- HTTP
- HTTPS
- FTP
- Fax
- Phone
- Other

For our example, simply provide a pointer to our web service, which has been deployed to `http://dotnetsvr/VBContactMgrWS/Contact.asmx`.

Finally, you need to publish Instance Info about your web service. This requires that you define a proper tModel (remember that a tModel is an interface to your web service). You need to first create a new tModel object in UDDI Services that will point to the WSDL interface generated for your web service by Visual Studio .NET. You can create a new tModel by right-clicking the tModels folder in the UDDI Services web interface and then choosing Add tModel. Let's name our tModel *Contact Manager tModel*, as shown in figure 8.24. Once your tModel is created, you have to assign a value to the Overview Document property. Notice that the Overview Document URL property points to the WSDL file of your web service. This process will properly define an interface for our tModel.



**Figure 8.24 Creating a tModel (using the Overview Document property)**

You can now create an Instance Info object. When you right-click your published binding and select Add Instance Info, you'll see a window that lets you search for a tModel. Your Instance Info will always point to an existing tModel, so let's perform a search for "Contact Manager" (our tModel is named Contact Manager tModel). When the results list appears, select the tModel that you want to use for your Instance Info object. Notice in figure 8.25 that we got one result from the search. When we click the "Contact Manager tModel" result, an Instance Info object is automatically created for our binding. You can use your new Instance Info object to describe parameters that are being passed to your web service function by either specifying the parameter names or the name of an HTTP-accessible file that describes them.

When you have finished this final step, your web service is now published in UDDI Services. At this point, you should test your work by performing various searches (or having someone else search for your service) in UDDI Services to ensure that all of the information you provided in the previous steps can be found and is intuitive. This step can be performed by using either the Windows Integrated Security web interface (<http://<serverName>/uddi>) or the public web interface (<http://<serverName>/uddipublic>).

When you have one of the web interfaces loaded into your browser, click Search to begin searching. You have the option of browsing categories or searching for services, providers, and tModels individually. You can also browse UDDI Services by using the Explorer in the leftmost frame of the page. In figure 8.26, we are searching for a service.

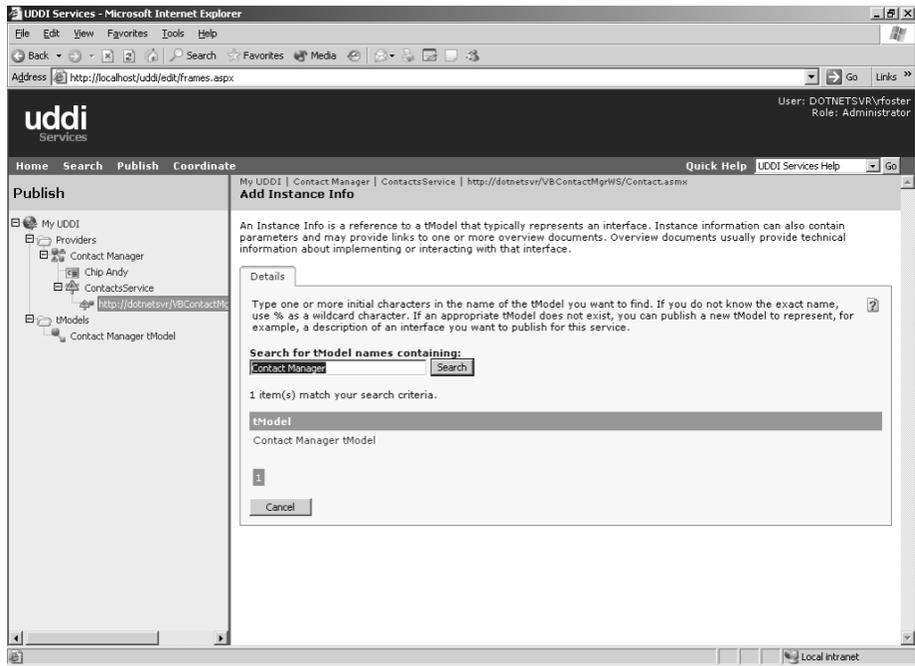


Figure 8.25 Publishing the Instance Info

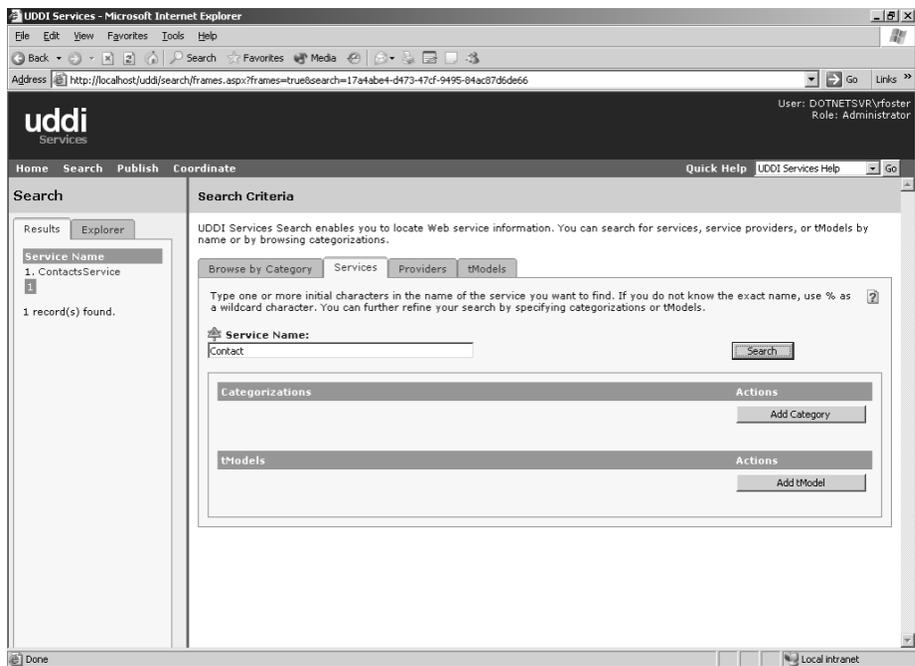
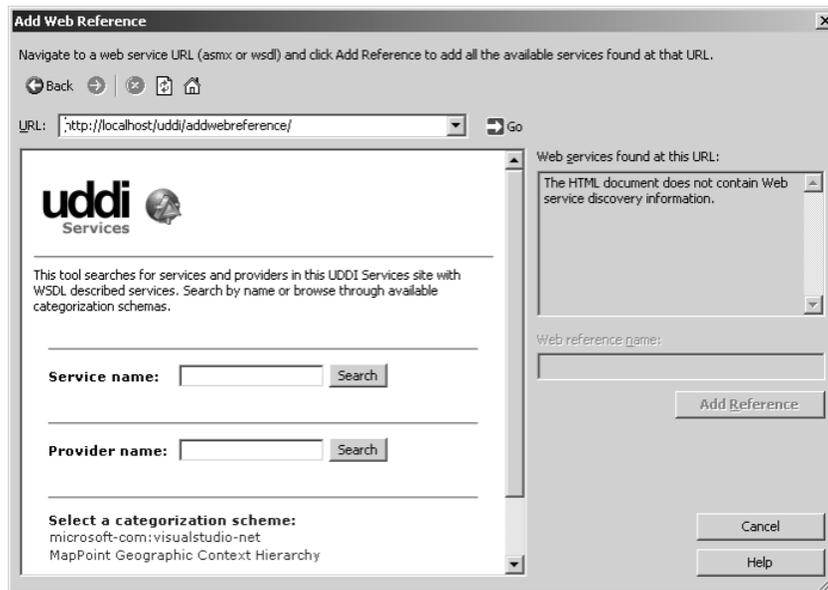


Figure 8.26 UDDI Services searching



**Figure 8.27** The UDDI Services Visual Studio .NET interface

Notice that you are required to specify only part of the word being searched; we specified *Contact* and the actual service name returned to the results pane in the leftmost frame is *ContactsService*. Remember that you should always test UDDI Services by searching for *all* information that you added in the previous steps in this section.

### **The Visual Studio .NET Add Web Reference interface**

UDDI Services can be used with Visual Studio .NET when you add a web reference to your project. This feature provides you with web service discovery inside Visual Studio .NET. You can access this interface from the Add Web Reference dialog box of your project, shown in figure 8.27.

Instead of typing an absolute URL to a web service, you can access UDDI Services from one of these URLs:

- `http://<serverName>/UDDI/addwebreference` (Windows Integrated Security interface)
- `http://<serverName>/uddipublic/addwebreference` (public interface)

The UDDI Services interface lets you search and discover locally on your network any web services that are registered in UDDI Services.

## **8.4 SUMMARY**

UDDI Services is a very powerful asset to your network infrastructure. If you are utilizing lots of web services in your applications and want to maximize reuse without compromising security, then you should consider UDDI Services.

In this chapter, we explained how to install UDDI Services (which is not installed by default when you set up Windows Server 2003). Then we discussed in detail the UDDI Services Console, which is used to manage UDDI Services' sites and servers. We looked at the properties that you can configure for each UDDI Services site and server and talked about which properties should be "tweaked" in order to maximize efficiency. Next, we published a web service to UDDI Services using the web interface and discussed the features of both the Windows Integrated Security web interface and the public web interface. Finally, we looked at how UDDI Services integrates with Visual Studio .NET to provide web service discovery.

In the next chapter, we will dive into securing your applications by looking at real-world security scenarios.