
How To Create Smart Tags

By creating your own Office XP Smart Tags you can improve the way that users communicate with others and use the software.

By Dennis Jarrett

Smart tags appeared first in Office XP applications, specifically the 2002 editions of Word, Excel, and Outlook. The technology is there both to improve editing and formatting techniques - essentially by offering appropriate choices when specific text was entered in a document - but also to facilitate intercommunication between applications. The example frequently quoted (which sadly doesn't always work) is a link between Outlook and Word; type a name and if that contact exists in an Outlook address book you should get the option of automatically inserting the mailing address or mailing or faxing to the name.

When the technology works, smart tags do allow Office applications to work together more efficiently. Rather than moving between multiple applications, smart tags can pass data from one application to another based on what the user is reading, reviewing or typing. Office XP also supports exposing smart tag actions within Internet Explorer (version 5.0 and up) so that documents created with Word or Excel will retain smart tags when viewed as a Web page.

The Black Art

Microsoft also included smart tags in beta versions of Internet Explorer 6, and here the feature was implemented in a rather different way, such that keywords on any Web site can be converted automatically into links to another site - without reference to or permission from the original publisher or author. Microsoft obviously intended to sell this service to recipient sites.

The legal basis for this is debatable. It's arguable that copyright is being infringed because an unauthorised derivative work is effectively being created by the deployment of the smart tags. Then there's a restraint-of-trade issue; publishers can argue that the use of smart tags may interfere with their ability to sell on-page advertising, since the smart tags themselves are effectively unauthorised ads.

Microsoft has dropped smart tags from Internet Explorer 6, at least for now. The facility may well return, since at least one official Microsoft spokesperson has been quoted as saying that "external feedback" was only one of the reasons for omitting smart tags from Internet Explorer 6.

In the meantime, get into the habit of including the metatag `<meta name="MSSmartTagsPreventParsing" content="TRUE">` in all pages you create. If Microsoft does implement smart tags in future releases of IE, this will prevent them being applied to your page.

If you use IIS, there is a simpler global solution that will disable smart tags for all served pages. Fire up the Internet Service Manager, right-click on the site's name and select Properties. Open the tab labelled http Headers, go to the Custom http Headers section, and click Add. In the pop-up window, enter MSSmartTagsPreventParsing in the Custom Header Name box. Enter TRUE in the Custom Header Value box. Click OK to create the new header, then click on Apply. Any page from this Web server will now have smart tags disabled.

Development And Deployment

Smart tags in Office are implemented via an API that provides dynamic recognition of text and provides relevant user actions. In practice, smart tags are deployed to the user's machine as a DLL component and/or an XML file.

A smart tag effectively consists of two lists. One is a list of words or phrases that it will recognise (called, predictably enough, recognisers). The other is a list of actions - also known as verbs - that the user will be invited to invoke when the recogniser finds something it recognises. When the smart tag engine spots a recogniser term, it applies a dotted purple underline to the text. If the user rests the cursor over the underlined text, a button will appear beside it. Click on the smart tag icon and an action menu pops up. This contains the custom-defined action list plus two standard entries (Smart Tag Options and Remove this Smart Tag).

To deploy smart tags you have four options. First, install an XML file on the user's computer. Second, register a DLL file on the user's computer. Third, embed individual smart tags in a particular source document (this applies only to Word or Excel documents - and of course the smart tag only works for that document). Fourth, embed individual smart tags in the HTML for a particular Web page (although you can label text in the Web pages as smart-tag actionable, nothing will actually happen unless a corresponding XML or DLL is installed on the user's computer).

In practice there are two basic ways to develop smart tags - as an XML file or via a registered DLL. The XML approach is quicker and simpler, and it's also generally neater if you need to update an existing smart tag. The drawback, however, is a killer: the only action possible with an XML smart tag is a clickthrough to a URL. For any other type of action you must use the DLL approach. Inevitably that's a lot more complicated, not least because it demands a thorough understanding of COM programming. Equally predictably, the DLL route can provide a lot more flexibility.

Developing A Smart Tag DLL

To develop a smart tag DLL, you'll need a COM development package such as Visual Basic 6. You will also need Office XP installed on the same box. It's also worth having Microsoft's Smart Tag SDK, which is a free download. Go to msdn.microsoft.com/office and click on the Smart Tag SDK link in the Essential Info box on the right.

The SDK is a tutorial, reference and sample code collection for developing smart tags for Office XP. There's no automation involved, though any competent user of Visual Basic or Visual C++ will be able to generate their own customised solutions quite easily with the information given. The SDK also supports XML and Office Web Components, so the whole kit provides enough information to develop smart tag recognisers in XML or HTML as well as DLL creation.

The Rules

Developing DLLs isn't too difficult, provided you follow all the rules. You use the

“Developing DLLs isn't too difficult, provided you follow all the rules. You use the Microsoft Smart Tags 1.0 type library, which is a collection of properties and methods grouped into two interfaces.”



Microsoft Smart Tags 1.0 type library, which is a collection of properties and methods grouped into two interfaces - one for recognisers, the other for actions. The DLL should include implementation code for all 18 of the properties and methods specified by these two interfaces. Once the DLL is complete it will have to be compiled and registered. Making manual changes to the registry is not recommended so you should consider using a tool such as the Microsoft Visual Studio Installer (msdn.microsoft.com/vstudio) to create a safe registration package that can also be distributed to other desktops.

One of the tutorials in the Smart Tag SDK shows you how to create a smart tag registration package by using the Visual Studio Installer, as well as registering subkeys manually for testing purposes.

Developing An XML File

Another useful download is the Advanced Smart Tag Tools bundle described at www.microsoft.com/OfficeDev/XP_06_2001/odc_stadvtools.htm. This does provide a degree of automation, at least for XML smart tag building. It includes a neat Excel worksheet template, the Smart Tag MOSTL Generator, which allows you to build smart tag XML list files. You also get the smart tag list builder, which is a DLL that allows you to build smart tag XML list files and list update files. Plus there's the MakeTrie and TestTrie tools, which allow you to build and test smart tag binary format files.

The MOSTL Generator, in particular, is a great first stab at automating smart tag production. It's quick and easy to - no coding is required, and it's sufficiently explicit for the not-too-technical end user to use. Essentially it generates an XML file which tells Office XP what terms to recognise and what actions to take. The other tool in this set, STTools, is a DLL (so you must of course run regsvr32.exe to register the DLL before you can use it) that contains a single COM object called STListBuilder. You can use it to create or edit one or both of the XML files for smart tag lists.

If you want to do it by hand, the example in Figure 1 shows the significant coding elements. It's a smart tag that pops up with a choice of online weather forecasts when the user types the word "Bristol". To use it [*the code is on your CD - Ed*], save the code in C:\Program Files\Common Files\Microsoft Shared\Smart Tag\Lists as weather.xml then double-click its icon. Now exit Office XP, load Word, type Bristol and see what happens.

The opening smarttaglist namespace declaration and the name element (a friendly name for this smart tag recogniser) are both mandatory. The description isn't, but it can obviously help in maintenance.

```
<FL:smarttaglist xmlns:FL="urn:schemas-microsoft-com:smarttags:list">
  <FL:name>weather</FL:name>
  <FL:description>Weather forecasts for Bristol, UK</FL:description>
  <FL:smarttag type="urn:schemas-microsoft-com:office:smarttags#City">
    <FL:caption>Weather</FL:caption>
    <FL:terms>
      <FL:termlist>Bristol,Avon</FL:termlist>
    </FL:terms>
    <FL:actions>
      <FL:action id="WeatherMet">
        <FL:caption>Weather forecast by Met Office</FL:caption>
        <FL:url> http://www.met-office.gov.uk/ukcity/bristol.html</FL:url>
      </FL:action>
      <FL:action id="WeatherBBC">
        <FL:caption>Weather forecast by BBC</FL:caption>
        <FL:url> http://www.bbc.co.uk/bristol/weather/overview.shtml</FL:url>
      </FL:action>
      <FL:action id="WeatherOnline">
        <FL:caption>Weather forecast by OnlineWeather</FL:caption>
        <FL:url>http://www.onlineweather.com/BritishIsles/Forecast/Bristol.html</FL:url>
      </FL:action>
    </FL:actions>
  </FL:smarttag>
</FL:smarttaglist>
```

“It’s easy enough to make a Web page smart-tag aware. Simply save a Word or Excel file as a Web page.”

Figure 1 - Sample XML smart tag.



The smarttag type is required. This is a unique XML namespace, specified as namespaceURI#tagname. You would normally create your own namespace, either within a DLL or as a URL reference to a server file. But it is possible to override XP's built-in naming - which is the option I took here. The namespace urn:schemas-microsoft-com:office:smarttags is the inbuilt XP namespace, and City is one of the tags for it. In fact you'll get the smart tag appearing when just about any city name is typed.

The terms element is required. The code in the example uses the child element termlist, a comma-delimited list of recognisers. The alternative is to specify a termfile here, this being the name and path of a specially-constructed .dat or .bin file that contains the terms. (If both termlist and termfile are specified, the former is used).

The actions list is also required. The action id is a required alphanumeric string that uniquely identifies the action in question; as specified by the following url element. The caption element identifies the action when the smart tag action list appears to the user.

Adding Smart Tags To Web Pages

It's easy enough to make a Web page smart-tag aware. Simply save a Word or Excel file as a Web page. Assuming the user has Office XP and the appropriate smart tag DLL or smart tag XML list description file installed, the smart tag Options button and associated actions will be appear on properly-tagged terms when the HTML is viewed with Internet Explorer.

You can also hand-code the smart tag, if you really need to. View the HTML source in a text or web page editor to see the structure. There's a useful and detailed explanation of what's involved at www.microsoft.com/officedev/xp_06_2001/odc_stwebpages.htm. In essence, however, your HTML coding needs to include: an Office namespace URI declaration; a smart tag namespace URI declaration for each smart tag type in the Web page; a smart tag type declaration for each smart tag type in the Web page; a smart tag OBJECT element; a smart tag element behaviour declaration for each smart tag namespace URI in the Web page. For every term you want to display, the Smart Tag Options button must be enclosed within the appropriate smart tag alias/tag name element.

“You can hand-code the smart tag, if you really need to. View the HTML source in a text or web page editor to see the structure.”

PCSA

Copyright ITP, 2001

New Reviews from [Tech Support Alert](#)

[Anti-Trojan Software Reviews](#)

A detailed review of six of the best anti trojan software programs. Two products were impressive with a clear gap between these and other contenders in their ability to detect and remove dangerous modern trojans.

[Inkjet Printer Cartridge Suppliers](#)

Everyone gets inundated by hundreds of ads for inkjet printer cartridges, all claiming to be the cheapest or best. But which vendor do you believe? Our editors decided to put them to the test by anonymously buying printer cartridges and testing them in our office inkjet printers. Many suppliers disappointed but we came up with several web sites that offer good quality cheap inkjet cartridges with impressive customer service.

[Windows Backup Software](#)

In this review we looked at 18 different backup software products for home or SOHO use. In the end we could only recommend six though only two were good enough to get our "Editor's Choice" award

[The 46 Best Freeware Programs](#)

There are many free utilities that perform as well or better than expensive commercial products. Our Editor Ian Richards picks out his selection of the very best freeware programs and he comes up with some real gems.