
Reading XML Data with a System Search

Robert Dickau

Senior Technical Content Manager
Flexera Software

Abstract

InstallShield 2008 introduces extensive XML functionality, including the ability to create and modify XML files on a user's system. This article describes functionality added to the System Search Wizard for Basic MSI projects to search for an XML file and read data from it.

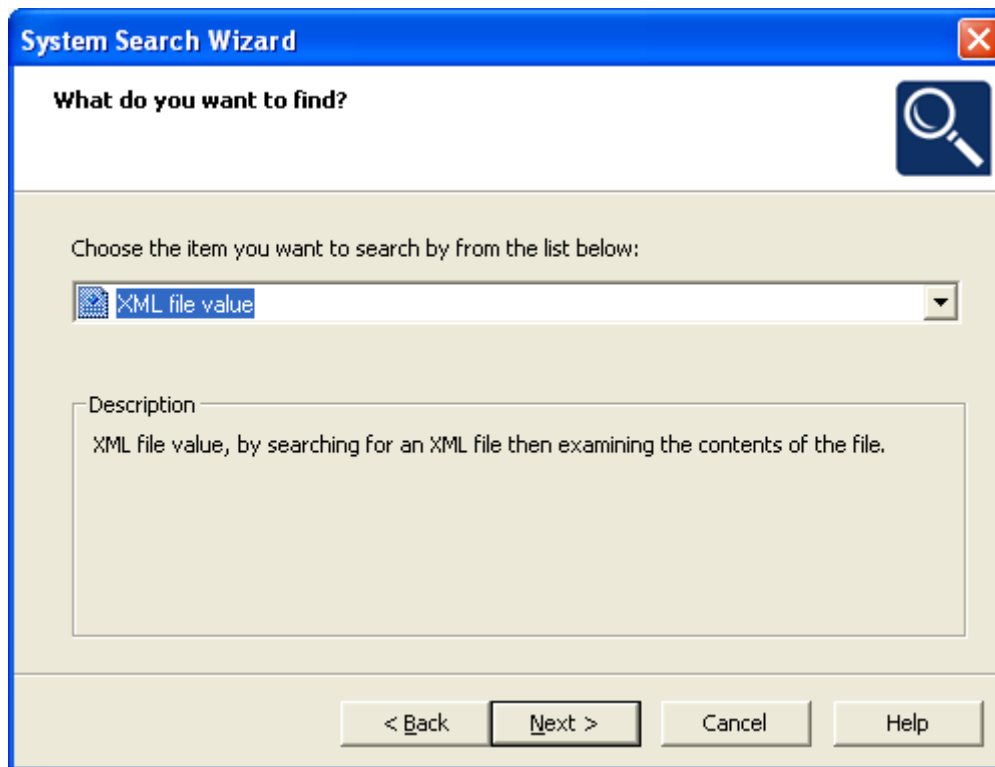
Setting Up the Search

To support the new Repositories feature, there is an XML file called **Repositories.xml** in the Repositories subdirectory of your InstallShield 10.5 distribution. Contents of this XML file might appear as follows.

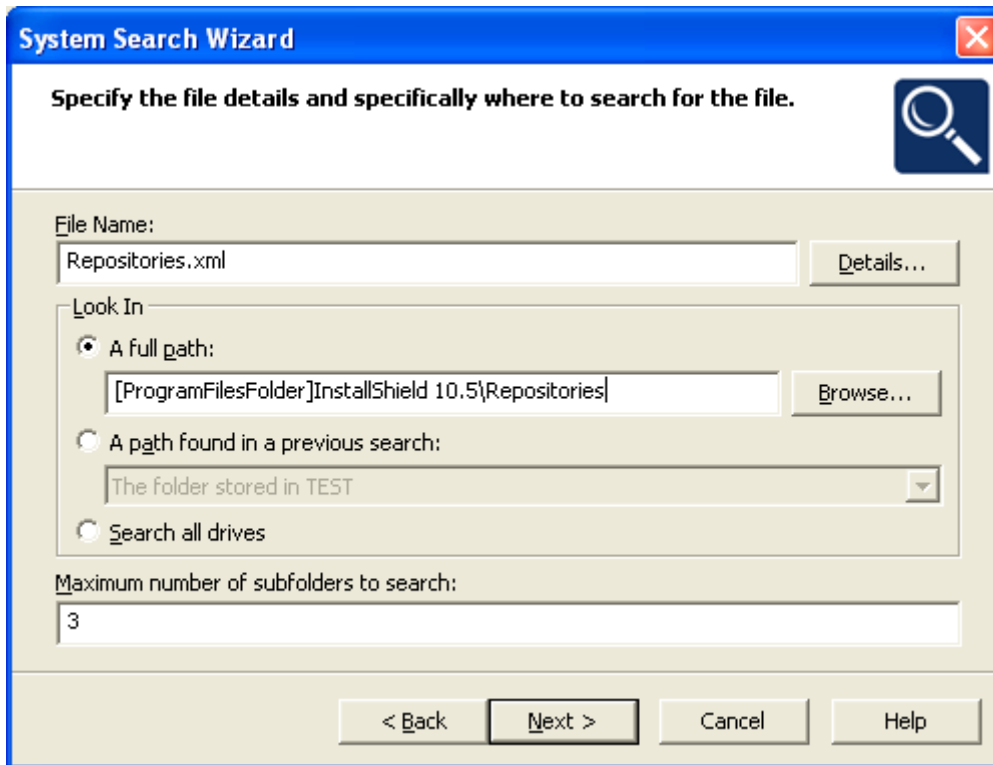
```
<?xml version="1.0" encoding="utf-8"?>
<Repositories>
  <Repository Key="Local" DisplayName="Local"
    Description="This is your repository where you can
      store elements on your system only."
    Path="C:\My InstallShield 10.5 Projects\LocalRepository\
      repo.xml" Local="1"/>
  <Repository Key="Network" DisplayName="Network Repository"
    Description="Exported project elements for use in
      multiple projects."
    Path="\\Server\Everybody\SharedElements\repo.xml" Local="0"/>
</Repositories>
```

For this example, suppose you want to read the Path attribute from the Repository element that has its DisplayName attribute set to "Local" (highlighted above).

To set up the search, launch the System Search Wizard by visiting the System Search view in the "Behavior and Logic" view group, right-clicking in the System Search list, and selecting Add. In the System Search Wizard, select the search type "XML file value".



In the Details panel, you specify the name and location of the XML file. When specifying the location of the file, you can click Browse to open the Browse for Directory panel, with which you can build up the search path based on existing Directory properties. In this case, the desired path is **[ProgramFilesFolder]InstallShield 10.5\Repositories**.

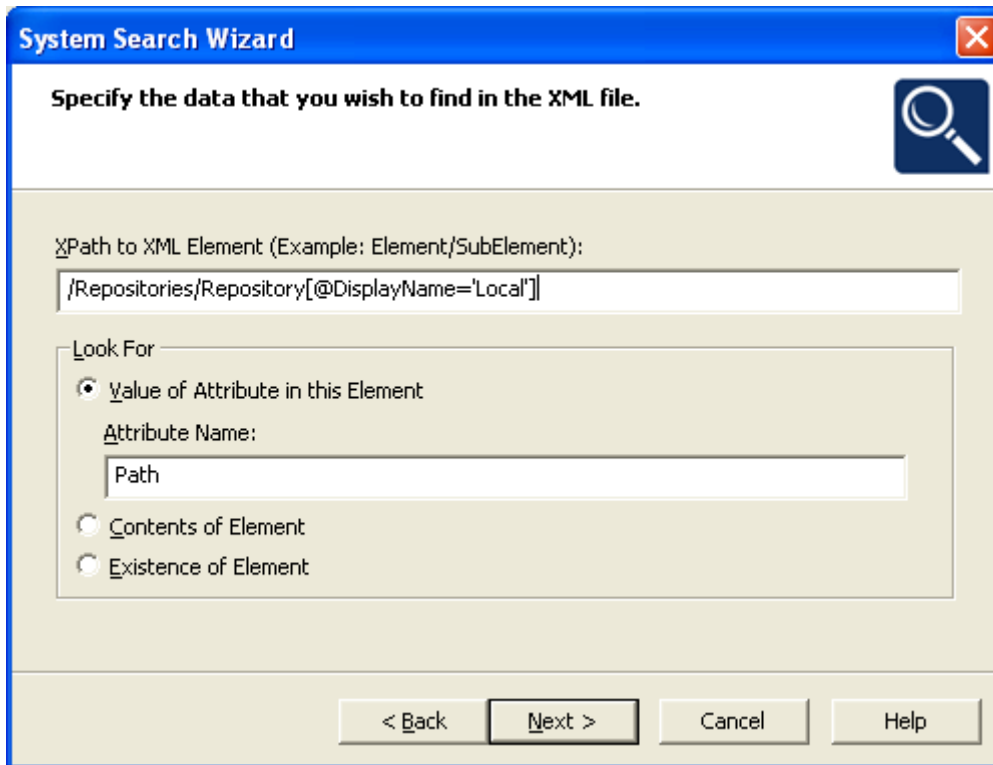


Next, in the Data panel, you specify the information you want to read from the XML file, if it is found. The data is specified using XPath syntax; for this example, the expression:

`/Repositories/Repository[@DisplayName='Local']`

indicates to look for the Repository subelement of the root Repositories element that has its DisplayName attribute equal to "Local".

Furthermore, in the Look For field you specify whether to read the value of an element's attribute, the element's contents, or just to detect whether the element exists. In this case, the search reads the value of the Path attribute of the detected element.



System Search Wizard

Specify the data that you wish to find in the XML file.

XPath to XML Element (Example: Element/SubElement):

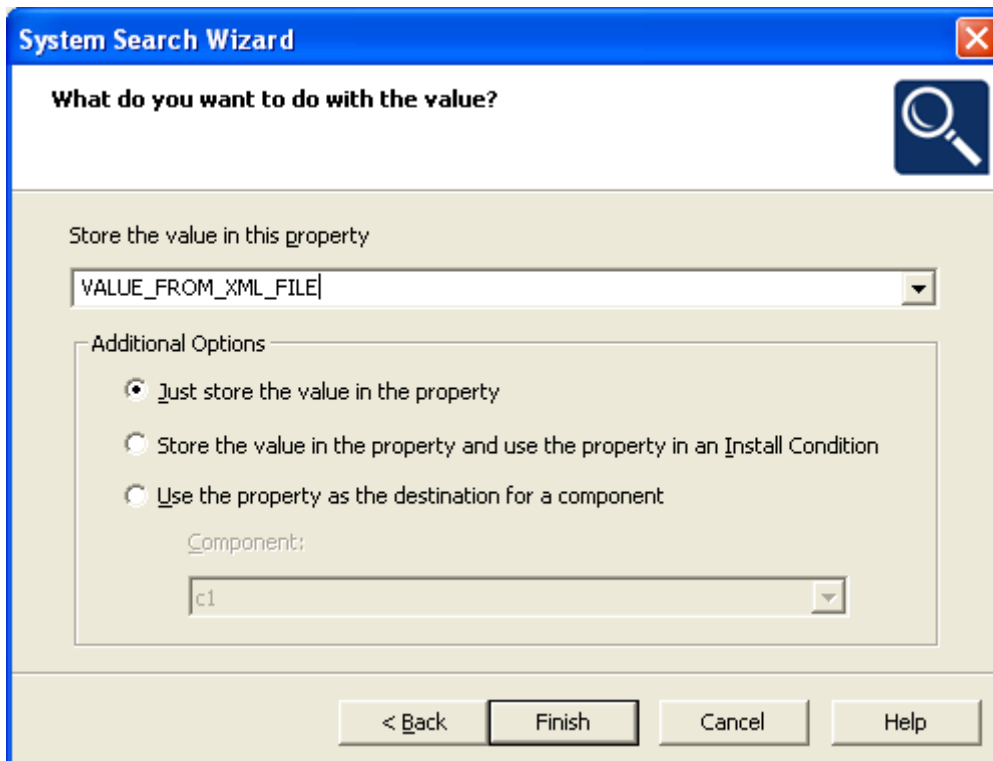
Look For

- Value of Attribute in this Element

Attribute Name:
- Contents of Element
- Existence of Element

< Back Next > Cancel Help

Finally, as with any system search, you specify the name of a public property in which to store the search results. In this case, the value of the specified attribute will be stored in a custom public property called VALUE_FROM_XML_FILE.



System Search Wizard

What do you want to do with the value?

Store the value in this property

Additional Options

- Just store the value in the property
- Store the value in the property and use the property in an Install Condition
- Use the property as the destination for a component

Component:

< Back Finish Cancel Help

You can use the value of the search property in any way appropriate for your project. To test the example, you can display the value of the search property on the InstallWelcome dialog box by adding a Text control with its Text value set to **VALUE_FROM_XML_FILE = [VALUE_FROM_XML_FILE]**.

At run time, if the search is successful, the InstallWelcome dialog box might appear as follows.

