

The Provider Keyword, ProgID, Versioning and COM CLSID Explained

Behind the Provider keyword is a chain of mappings to the software implementation contained in a dll file. This article explains that chain and how COM works in general.

The connection string **Provider** keyword specifies what OLE DB provider to use. To set this to an OLE DB provider, specify the value for the Provider keyword to the 'VersionIndependentProgID' value specific to the provider.

For example,

```
Provider=MSDASQL;  
// or  
Provider=Microsoft.Jet.OLEDB;
```

Where **MSDASQL** and **Microsoft.Jet.OLEDB** are examples of 'VersionIndependentProgID' of an OLE DB provider.

The value can also be set to the 'ProgID' of the provider, which may have a version attached to it. It is not 'Version Independent'. It is 'Version Specific'.

For example,

```
Provider=MSDASQL.1;  
// or  
Provider=Microsoft.Jet.OLEDB.4.0;
```

Where **MSDASQL.1** and **Microsoft.Jet.OLEDB.4.0** are examples of (version **specific**) 'ProgID' of an OLE DB provider.

Why ProgID with a version?

If two versions of a provider are installed on the same system, use the 'ProgID' to specify exactly which version to use. If two versions are installed on a system and the 'VersionIndependentProgID' value is specified, the **most recent** version of the OLE DB provider will be used.

OLE DB and ODBC

If no Provider keyword is specified in the string, the **OLE DB Provider for ODBC** (MSDASQL) will be the default value. This is because of backwards compatibility with ODBC connection strings. The ODBC

connection string in the following example can be passed in, and it will successfully connect.

```
Driver={SQL Server};Server=localhost;Trusted_Connection=Yes;Database=myDb;
```

So the above example will use the **OLE DB Provider for ODBC**, shorthand MSDASQL which is the default OLE DB Provider, and start the connection with an **Microsoft SQL Server ODBC Driver** ({SQL Server}) loaded to the local computer SQL Server instance. This would therefor be the same as specifying:

```
Provider=MSDASQL;Driver={SQL Server};Server=localhost;Trusted_Connection=Yes;Database=myDb;
```

So, what is a ProgID anyway?

The ProgID is a alias for a COM component registration. For instance the **Microsoft.ACE.OLEDB.12.0** OLE DB provider is contained in the file **ACEOLEDB.DLL** and it is registered on the computer as a COM component. The registration of a COM component is made in the computers registry using a CLSID which is a globally unique identifier that identifies a COM class object. '{dee35070-506b-11cf-b1aa-00aa00b8de95}' is an example of a CLSID.

Instead of referring to a COM component, such as the **Microsoft.Jet.OLEDB.4.0** OLE DB provider, through it's CLSID, there is also a mapping from a more friendlier name, the ProgID, to the CLSID. This PROGID - CLSID mapping is also stored in the registry of the computer.

If two different versions of a COM component is installed (there are two different dll files containing the provider software) they will have two different CLSIDs in the registry and they will have two different ProgIDs mapped. If we just want to load the latest version on the system we can use the 'VersionIndependentProgID' instead which is another registry entry on the local computer pointing to the latest version of the registered COM components CLSID. For instance the 'Microsoft.Jet.OLEDB' value is the Version Independent ProgID for the JET OLE DB provider.