



# IBM Software Technical Document

## Document Information

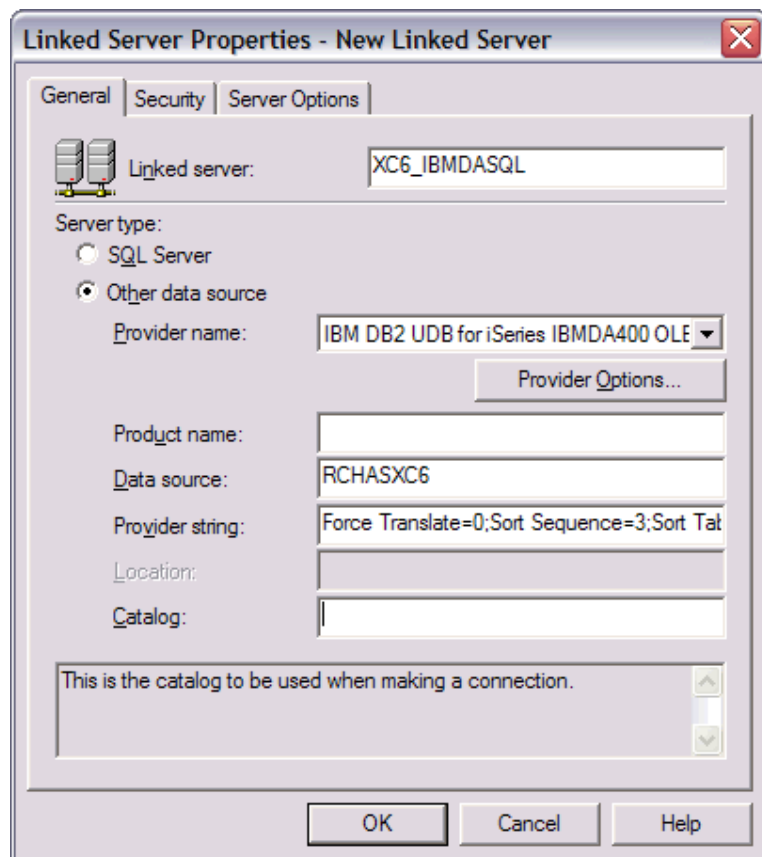
Document Number: **404456535**  
 Functional Area: Client Access  
 Subfunctional Area: OLE DB  
 Sub-Subfunctional Area: General  
 OS/400 Release: V5R3M0; V5R4M0  
 Product: IBM ISERIES CLIENT ACCESS EXP (5722XE100)  
 Product Release: N/A

## Document Title

**Sort Order and SQL Server Linked Servers**

## Document Description

Microsoft's SQL Server product is often used to link to DB2 UDB for iSeries as a linked server. This connection is an OLEDB consumer so it can be made using the OLEDB providers shipped with iSeries Access (IBMDA400 or IBMDASQL) or it can use the iSeries Access ODBC driver through Microsoft's OLEDB provider for ODBC Data Sources (MSDASQL). When the linked server is created, the connection properties can be defined on the provider string. The special properties for the iSeries Access OLEDB providers may be specified here, separating each property from the next using semicolons. An example might look similar to the following:



In this case, the linked server XC6\_IBMDASQL is being created. The IBMDASQL provider is selected as the OLEDB provider. The system name is used as the Data Source. In this example, the Provider string is specifying the properties, "**Force Translate=0;Sort Sequence=3;Sort Table=MIKSWENS/EN\_US\_SW**". Refer to the following Rochester Support Center knowledgebase document for a complete listing of all of the custom properties available for the iSeries Access OLEDB providers:

23062121, **iSeries Access for Windows Custom Connection Properties for OLE DB:** 

The ones specified here force conversion of character data stored in CCSID 65535 and will use a custom sort sequence table, MIKSWENS/EN\_US\_SW that was created to provide a custom sort order.

Queries are performed against the linked servers using one of two syntaxes. The first one uses a 4-part naming convention that consists of the linked server name, the database or catalog name, the collection (library), and table or view (file). An example using the server defined above looks similar to the following:

```
SELECT TEXT1 FROM XC6_IBMDASQL.RCHASXC6.MIKSWENS.UNITEST ORDER BY TEXT1
```

This should look familiar to anyone with a passing knowledge of SQL. In this case though it will produce incorrect output, the results will not be sorted as they should be. The reason for this is that SQL Server does not pass this query directly on to the OLEDB provider. Ultimately, it executes the select statement and performs the sort using default sort order that was defined during the install of SQL Server. Although this syntax is more familiar, it does not provide correct output.

The second syntax uses looks more like two queries, a select of the data returned by a passthrough query. The query equivalent to the one above looks like the following:

```
SELECT * FROM OPENQUERY ( XC6_IBMDASQL, 'SELECT TEXT1 FROM MIKSWENS.UNITEST ORDER BY TEXT1' )
```

This second query actually performs better than the first because SQL Server makes fewer communications flows to the server. When the first syntax is used, SQL Server runs the statement several times so that it can retrieve metadata before it finally runs a version of the statement from which it will retrieve results.

PMR Number:	<input type="text"/>	
Related APARs:	<input type="text"/>	
Related Public Documents:	<input type="text"/>	
	<input type="text"/>	
	<input type="text"/>	
	<input type="text"/>	

IBM disclaims all warranties, whether express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. By furnishing this document, IBM grants no licenses to any related patents or copyrights. Copyright © 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009 IBM Corporation. Any trademarks and product or brand names referenced in this document are the property of their respective owners. Consult the [Terms of use](#) link for trademark information.