

How to write "Passthrough" SQL statements and "Range Where" clauses in Microsoft Great Plains Dexterity

CONFIDENTIAL ARTICLE

(The information in this article is provided to you in accordance with your Confidentiality Agreement)

INTRODUCTION

This article describes how to write **Passthrough** SQL statements and **Range Where** clauses in Microsoft Business Solutions - Great Plains Dexterity.

MORE INFORMATION

When you write **Passthrough** SQL statements and **Range Where** clauses in Dexterity, make sure that the Transact SQL code is compatible with all regional settings and sort orders. Also, make sure that field values that are passed to Microsoft SQL Server are handled correctly. To do this, use the following methods:

- If you want to pass a date value to Microsoft SQL Server, use the Dexterity **sqlDate()** global function to make sure that the date is in YYYYMMDD format. If you use the Dexterity **str()** function, the program uses either the MM/DD/YYYY format or the DD/MM/YYYY format, depending on the regional settings. SQL Server does not accept the DD/MM/YYYY format because this format may cause intermittent errors that occur after the twelfth day of each month.

Note The **sqlDate** function does not add the single quotation marks that are required when you pass the date to SQL Server. Therefore, enclose the result of the function by using single quotation marks.

- If you want to pass a string value to SQL Server, use the Dexterity **SQL_FormatStrings()** global function to make sure that strings are wrapped in single quotation marks. If a single quotation mark exists in the string, you must add a second single quotation mark to stop the premature termination of the string.
- When a range is selected between the minimum and maximum values on a string field, the maximum value that is used by Dexterity may not be the correct maximum value for SQL Server. The correct maximum value for SQL Server depends on the sort order and the collation of the instance of SQL Server. By using the Dexterity **fill** command, the program fills the field by using ASCII 255 for the length of the string. However, depending on the sort order, ASCII 255 may not be the highest value. Therefore, the **where** clause will return no results. The maximum value for ASCII 255 is \ddot{y} (lowercase y with umlaut). This character is treated as a Y by some SQL sort orders. This means that the end of the range is set to Y in MSSQL and that anything that starts with the character Z is not included in the range. The following Dexterity code is an example that uses the **system 9600** command to obtain the maximum character value for a system's sort order. The code then uses this value to build a **where** clause.

```
local string      l_MaxChar;
local integer     l_Result,l_Length;

system 9600, table SOP_HDR_WORK, l_MaxChar, l_Result;

if empty('End Location Code') or filled('End Location Code') then
    fill 'End Location Code';
    l_Length = length('End Location Code');
    clear 'End Location Code';
```

Partner Only Article

Article ID : 910129
Last Review : N/A
Revision : 2.1

```
'End Location Code' = pad('End Location Code', TRAILING, l_MaxChar, l_Length);  
end if;  
  
range table SOP_HDR_WORK where physicalname('Location Code' of table SOP_HDR_WORK) + CH_SPACE + CH_GREATERTHAN + CH_EQUAL + CH_SPACE  
+ SQL_FormatStrings('Start Location Code') + CH_SPACE + SQL_AND + CH_SPACE  
+ physicalname('Location Code' of table SOP_HDR_WORK) + CH_SPACE + CH_LESSTHAN + CH_EQUAL + CH_SPACE  
+ SQL_FormatStrings('End Location Code');
```

APPLIES TO

- Dexterity, when used with:
Microsoft Dynamics GP 9.0
Microsoft Business Solutions–Great Plains 8.0

Keywords: kbmbspartner kbmbsmigrate kbhowto KB910129