

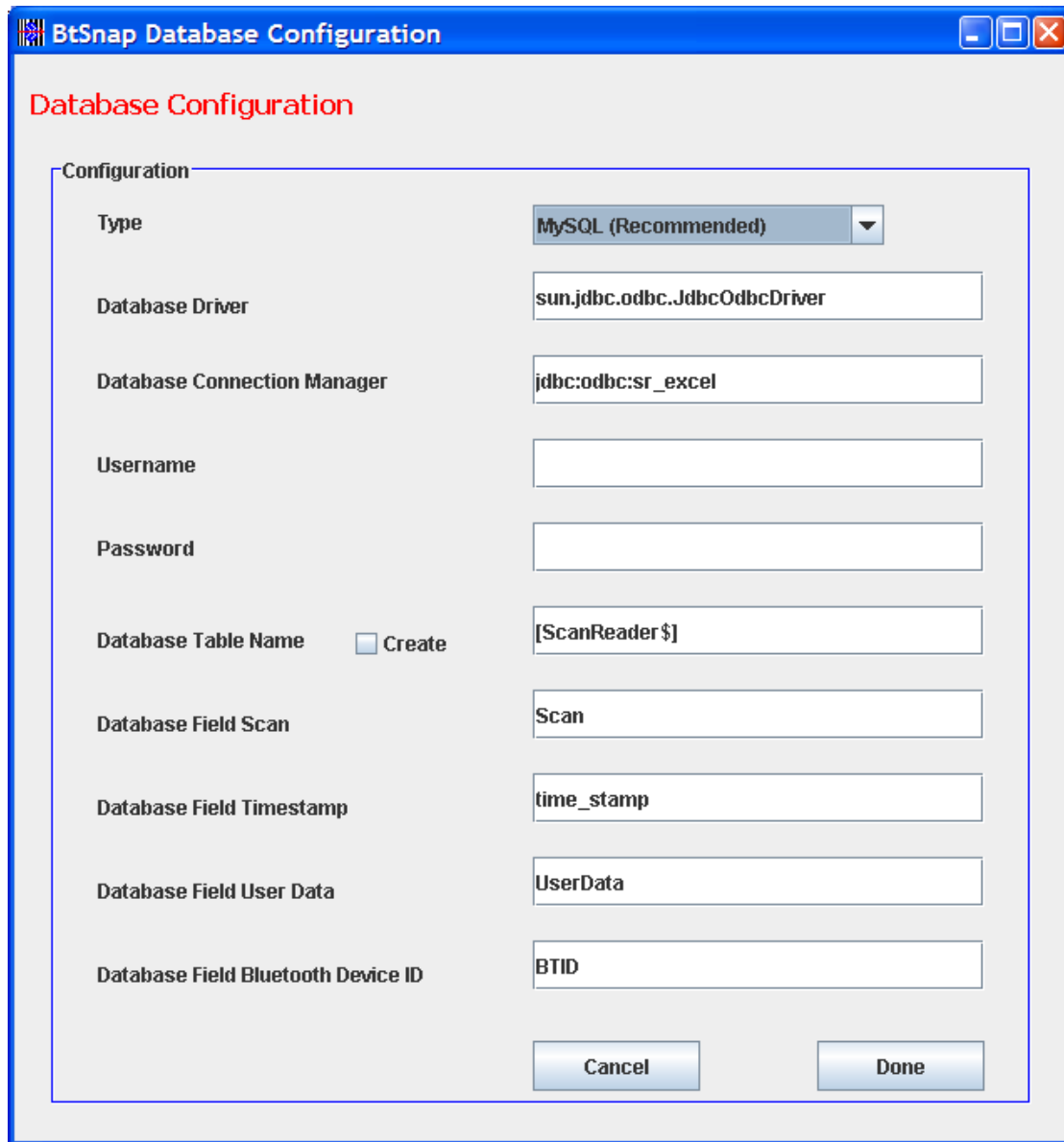
## Using ScanChamp® ScanReader with Excel

This document outlines using ScanChamp® ScanReader (Solo and BtSnap versions) to interface directly to Excel via ODBC. When interfacing with Excel via ODBC some additional considerations need to be made other than when using a traditional relational database such as MySQL, Oracle, SQL 2000, or Access.

This example assumes a copy of Microsoft Excel is running on Windows XP on the same machine.

1. Create an Excel file (for example, "C:\ScanReaderScans.xls").
2. Name the sheet in the file you plan on storing scanned data into whatever you want (you can use Sheet1, but just make a note of what the name is). For this example, we will call the sheet "ScanReaderScans".
3. Create an ODBC connection. To do this, go into Administrative Tools, Data Sources (ODBC). Under the "User DSN" tab, click "Add...". Select "Microsoft Excel Driver (\*.xls)". Click "Finish". It will now ask you to provide a "Data Source Name" as well as a "Description". The description is not important, but make note of the Data Source Name. For this example, it will be called "sr\_excel". You also need to specify the file you want to use under "Select Workbook". Additionally, click the "Options>>" tab and uncheck "Read Only" or data won't be able to be scanned into the worksheet.
4. After the Excel ODBC Driver is in place, one more item is required before sending data from the BtSnap to the Excel spreadsheet. Open up the Excel spreadsheet that will be used (in this example "C:\ScanReaderScans.xls"). Then modify the sheet that by adding the appropriate column names. These will be the field names to be used in ScanChamp® ScanReader ODBC configuration table. For example, we will add "Scan", "time\_stamp", "UserData", and "BTID" for columns 1 to 4, respectively. At this point you should have a blank Excel file with column names at the top. Make sure all of the field titles are text fields.
5. Start ScanChamp® ScanReader. Under "Database Configuration", set the fields according to the names entered into the Excel document (see screenshot below). Notice the following fields are set as follows (be sure you have "create" unchecked):

- **Type** – This can be left as MySQL.
- **Database Driver** – “sun.jdbc.odbc.JdbcOdbcDriver”
- **Database Connection Manager** – “jdbc:odbc:sr\_excel” Note the Data Source Name you used when you created the Microsoft Excel Driver. For yours, you will put “jdbc:odbc:YourMicrosoftExcelDriverName”.
- **Username** – Leave Blank.
- **Password** – Leave Blank.
- **Database Table Name** – “[ScanReaderScans\$]” This is probably the most important and confusing part. You will use the sheet name from your Excel file. In my case I called it ScanReaderScans. You need to place brackets around the entire name, as well as a ‘\$’ after the sheet name. This is because of the weird way in which the Microsoft Excel Driver handles table names.
- **Database Field Scan** – “Scan”. Alternatively, use whatever you called the Scan field from step 4 above.
- **Database Field Timestamp** – “time\_stamp”. Same as above.
- **Database Field User Data** – “UserData”. Same as above.
- **Database Field Bluetooth Device ID** – “BTID”. Same as above.



**Database Configuration**

Configuration

Type: MySQL (Recommended)

Database Driver: sun.jdbc.odbc.JdbcOdbcDriver

Database Connection Manager: jdbc:odbc:sr\_excel

Username:

Password:

Database Table Name:  Create [ScanReader\$]

Database Field Scan: Scan

Database Field Timestamp: time\_stamp

Database Field User Data: UserData

Database Field Bluetooth Device ID: BTID

Buttons: Cancel, Done

Now when a read is performed with ScanReader Solo ScanReader for BtSnap, ScanChamp® ScanReader software will insert the download scans into the Excel File that has been configured to receive them.

Note: In Excel, rows that have had data in them are considered “used rows”, even after the “Clear” function has been used. ScanChamp ScanReader Solo will not write data to these rows, but will find the next unused row in the spreadsheet. You **must “Delete” rows that have ever had data in them** if you want to write a new set of data to that row.