

Creating Custom Reports

Your DaySmart software, i.e. Salon Iris, Orchid, 123Pet, and InkBook, allows you to create your own custom reports. There is no limit to the number of reports you can create. Each time you create a new report you have the option of having it added to the **Reports** drop-down menu.

You can even create your custom reports by basing them on the built-in reports that come with Salon Iris. See the **Creating Reports Based on Built-In Software Reports** section below.

Creating Your Own Reports

Creating a custom report is not difficult; however, you need to be familiar with the following two topics:

1. **The software database format:** Your database is where the software holds all your information. It contains a list of all your clients, products, services, sales, etc. These items are grouped together in separate lists known as tables. The tables are further subdivided into smaller categories called fields. For example, the **Clients** table has a field called **FirstName**, which is a list of all the clients' first names. You can see a complete list of the database format by opening the file called **Database Schema.xls** located on the software installation CD.
2. **Structured Query Language:** The Structured Query Language, or SQL, is a series of commands you use to specify what information you want in your report. The format consists of abbreviations and words similar to English.

Perform the following steps to create a custom report:

1. From the **Reports** drop-down menu, select **Manage Custom Reports**.
2. Click the **Add** button and type in the title, description, and SQL statement.
3. See the examples in the **Structured Query Language Introduction** section shown below.
4. Click the **Run** or the **OK** button.

Creating Reports Based on Built-In Software Reports

You can view the SQL statements for approximately 90% of the built-in reports in your software. Importing the built-in reports in your software is a great way to see how the reports were written. You can model your own reports after these built-in reports. To view the SQL statements for an existing report and create your own report, perform the following steps:

1. From the **Reports** drop-down menu, select **Manage Custom Reports**. The **Custom Reports** screen will appear.
2. Click the **Add** button.
3. Click the **Import** button. The **Available Reports** screen will appear.

4. Select the report you wish to import the SQL statements for and press **OK**.
5. Make the desired changes to the SQL statements: title and description, and then click **OK** to return to the **Custom Reports** screen.
6. Click **OK** again and your report will be added to the list of custom reports.
7. You can view the custom report that you just wrote by going to the **Reports** drop-down menu and selecting **Manage Custom Reports**. The **Custom Reports** screen will appear.
8. Select your report and click the **Run** button. Click **OK** when finished.

Structured Query Language Introduction

Below is a basic introduction to the SQL language. This introduction will allow you to write basic reports. SQL statements can range from very simple to extremely complex. Because SQL is one of the most popular database languages, hundreds of books have been written on it. We highly recommend purchasing a book on SQL if you would like to become more familiar with the language. After you have a firm understanding of SQL, you will be able to write your own very sophisticated reports.

The basic format of an SQL statement is as follows:

- **SELECT¹ <field> FROM <table>**
 - **Where:**
 - <field> are a list of the items, or fields, you want to view, or * if you would like to view all fields.
 - <table> is the name of the table containing the items.
- **Example 1: List of All Clients**

As we mentioned above, there is a table in the database called **Clients** that contains an item, or field, called **FirstName**. A SQL statement to obtain a list of all the first names of all your clients would be:

```
SELECT fldFirstName2 FROM tblClients
```

We could also obtain a list of all the information on each client with the following statement:

```
SELECT * FROM tblClients
```

1: It is important that the command words, i.e. SELECT, FROM, etc., be capitalized.

2: Notice that all fields are preceded by **fld** and all tables are preceded by **tbl**. This is true for all fields and tables in your DaySmart software.

1. To create a custom report that shows a list of all the clients, select **Manage Custom Reports** from the **Reports** drop-down menu. Click the **Add** button, and enter the following information:
 - **Title:** **List of All My Clients**
 - **Description:** **This is my first custom report**
 - **SQL Statement:** **SELECT * FROM tblClients**
2. Press the **Run** button. You should see a list of all your clients and all their information. Notice that each field name, the heading of each column, is preceded by **fld**. We will show you how to clean this up in the next example.
3. Click **OK**.
4. Click **OK** again.

- **Example 2: Better List of All Clients**

Example 1 showed you how to get a basic list of all clients and all their information. The result was a list that was very complete, but also very hard to read as the column headings were all cryptic and preceded with a 'fld.' You can indicate your own column heading names by using the **AS** command. You can also specify what fields you want displayed. In the example below you can type anything you want in between the brackets: **[Column Heading]**. Whatever is typed in between the brackets is your new column heading.

Here is how to create a list of only the first and last name of each client, complete with easy-to-read column headings:

- **Title:** **First and Last Name of All My Clients**
- **Description:** **This report is easier to read**
- **SQL Statement:** **SELECT fldFirstName AS [First Name], fldLastName AS [Last Name]
FROM tblClients**

- **Example 3: Show Only Certain Clients**

Now let's make a more useful report. We will create a report that contains a list of all clients who have a first name of Chris and have spent more than \$100. We will use the **WHERE** command to do this. The **WHERE** command is placed after the name of the table and is used to select only entries matching a specific criteria.

- **Title:** **All Chris's That Spend More Than \$100**
- **Description:** **These clients have a first name of Chris and spent more than \$100**
- **SQL Statement:** **SELECT fldFirstName AS [First Name], fldLastName AS [Last Name]
FROM tblClients WHERE fldFirstName='Chris' AND fldTotal>100**

The above examples are only an introduction to SQL. The language is very sophisticated and supports dozens of other keywords. Over 90% of the reports in your software were written using pure SQL. You can view all the SQL statements by pressing the **Import** button when editing a report, then selecting the built-in report that contains the SQL statement you wish to view.

Built-in User Prompts

There will be times when you want to ask the user for information when you are writing a report. In addition to all SQL commands, your software allows you to imbed additional commands in your SQL statements. They are as follows:

- **GetUserInput(<prompt>)**
 - **Where:**
 - <prompt> is the question you want to display to the user
- **GetUserData(<type>,<field name>,<prompt>)**
 - **Where:**
 - <type> is WHERE or AND
 - <field name> is the name of the field you want to compare the selected date to
 - <prompt> is the question you want to display to the user

When this keyword is embedded in a SQL statement, the user will be prompted to enter a value. The value is inserted into the original SQL statement in place of the **GetUserInput** keyword.

- **Example:**

```
SELECT * FROM tblClients WHERE fldFirstName='GetUserInput ('Enter the client's first name.')
```

When this report is run, the user will be prompted with a pop-up window with the prompt **Enter the client's first name**. If the user enters **Bill**, the final SQL statement would be:

```
SELECT * FROM tblClients WHERE fldFirstName='Bill'
```

GetUserData(<type>,<field name>,<prompt>)

When this keyword is embedded in the SQL statement, the user will be prompted to enter a range of dates. The range of dates selected by the user is transformed into another SQL statement using the <field name> that is passed. The result is inserted into the original SQL statement in place of the **GetUserData** keyword.

- **Example 1:**

```
SELECT * FROM tblClients GetUserData ('WHERE','fldLastVisit','Enter last visit date range')
```

When this report is run the user will be prompted with a pop-up window and asked to enter a date or range of dates. In this example, to select a date range from Jan 1, 2010 to Jan 1, 2011, the final SQL statement would be:

```
SELECT * FROM tblClients WHERE fldLastVisit >= DateSerial (2010,1,1) AND fldLastVisit <= DateSerial(2011,1,1)
```

- **Example 2:**

If we wanted to specify additional search criteria, we could place it in prior to the **GetUserData** keyword then use the **AND** type in the **GetUserData** keyword. For example:

SELECT * FROM tblClients WHERE fldFirstName='Chris' GetUserData('AND', 'fldLastVisit', 'Enter last visit date range')

When this report is run the user will be prompted with a pop-up window and asked to enter a date or range of dates. In this example, to select a date range from Jan 1, 2010 to Jan 1, 2011, the final SQL statement would be:

SELECT * FROM tblClients WHERE fldFirstName='Chris' AND fldLastVisit >= DateSerial(2010,1,1) AND fldLastVisit <= DateSerial(2011,1,1)