How To Guide

PUBLIC

SAP Business One 10.0 and SAP Business One 10.0, version for SAP HANA

Document Version: 1.2 – 2019-10-16

How to Define and Use User-Defined Values

Typographic Conventions

Type Style	Description
Example	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
Example	Emphasized words or expressions.
EXAMPLE	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE	Keys on the keyboard, for example, F2 or ENTER.

Document History

Version	Date	Change
1.0	2009-10-09	First version
1.1	2019-01-03	Minor updates for SAP Business One 9.3.
1.2	2019-10-16	Add Multiple Trigger feature in Auto Refresh section for SAP Business One 10.0.

Table of Contents

1 Introduction

SAP Business One makes filling out documents and entering data into forms easier by enabling you to set up user-defined values, that is, your own valid values for different fields. For a specific field, you can use user-defined values to do the following:

- Display a list of values from which the user can choose.

  The list of values can be static or can be derived from a query that is run each time the user edits the field.
- Automatically fill in a field whenever another field (the trigger field) is changed. For example, you can automatically fill in the correct Zip code based on the address that was entered.

User-defined values can be set up for both system and user-defined fields.

> **Note**
>
> The user-defined values feature does not necessarily require the user to select from the defined values. The user can still manually enter or change a value – whether it was entered automatically or selected by the user from a dropdown list.

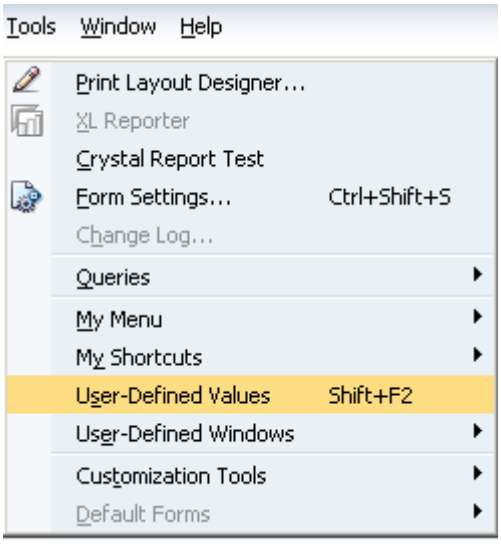
2 Using the Menu Bar

This section introduces you to the basic functions for user-defined values that you can access through the menu bar.

2.1 Activating User-Defined Values for a Field

To see user-defined values for a field, you need to activate them as follows:

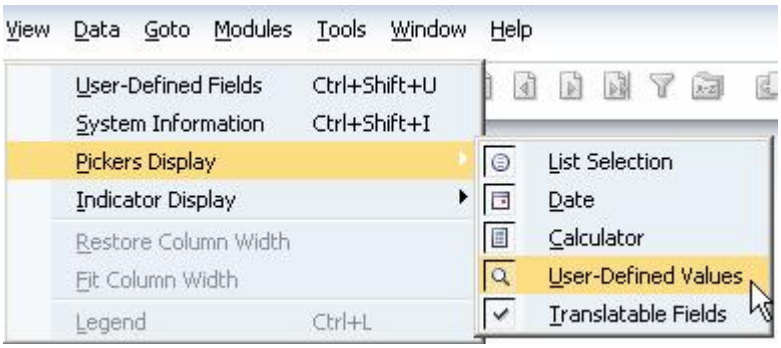
1. Place your cursor in the field and do one of the following:
   - Choose *Tools* ® *User-Defined Values*.
   - Press `Shift`+`F2`.



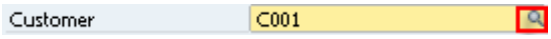
2. If a user-defined value has not been assigned to the field, click *Yes* in the *System Message* window.

2.2 Displaying the Icon in Fields with User-Defined Values

If you want to find out which field have user-defined values in a window, choose *View* ® *Pickers Display* ® *User-Defined Values*.



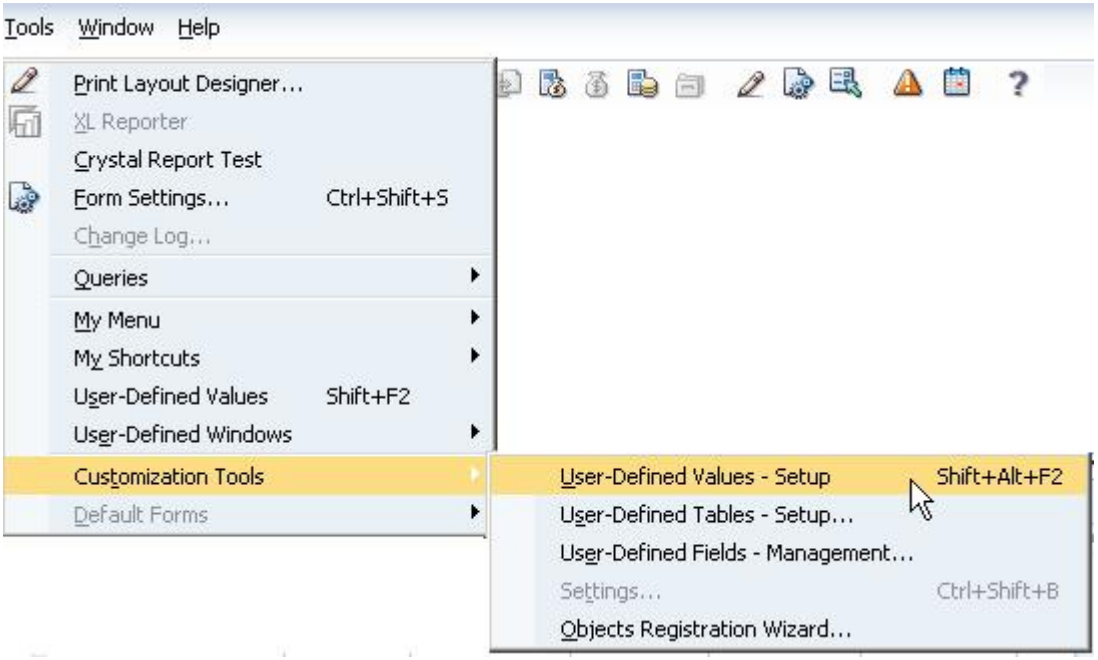
The user-defined values icon appears next to each field for which user-defined values have been set.



2.3 Setting Up a User-Defined Value for a Field

Place your cursor in the field and do one of the following:

- Choose *Tools* ® *Customization Tools* ® *User-Defined Values – Setup*.
- Press `Alt`+`Shift`+`F2`.



For more information, see *Setting Up User-Defined Values*.

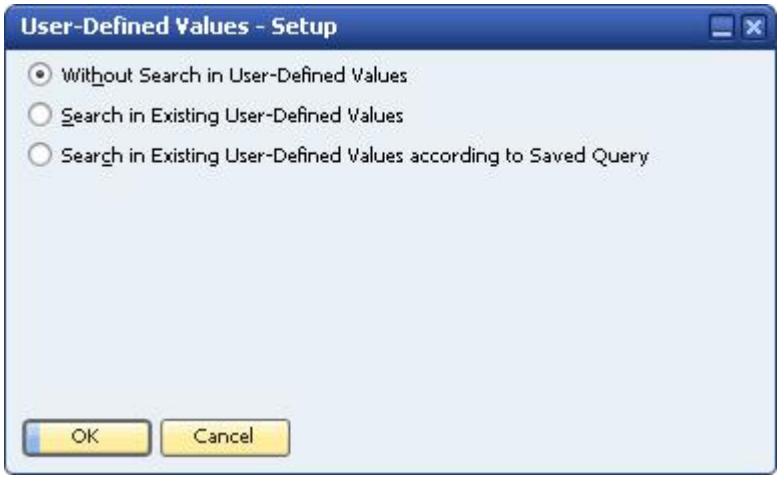
3 Setting Up User-Defined Values

This section describes how to set up user-defined values for a field.

Procedure

1. Place the cursor in the field for which you want to define values, and press Alt+Shift+F2. Alternatively, you can place the cursor in the field for which you want to define values and choose from the menu bar *Tools* ® *Customization Tools* ® *User-Defined Values - Setup*.

   The *User-Defined Values - Setup* window opens:



2. Choose one of the following options:

Option	Description	Next Steps
Without Search in User-Defined Values	Use this option to cancel previously defined user-defined values. If a static list of predefined values was created, the values are no longer relevant but remain in the system. If you select the *Search in Existing User-Defined Values* option again, the previously-defined values are available. If a query was selected to provide user-defined values, the reference to the query and related settings are removed.	
Search in Existing User-Defined Values	Enables the user to select from a static list of predefined values.	After selecting this option, follow the instructions in *Setting Up a List of Predefined Values*.

Option	Description	Next Steps
Search in Existing User-Defined Values according to Saved Query	Enables one of the following: · The user to select from a list of values derived from a query. · The system to set the field value automatically based on the result of a query.	After selecting this option, follow the instructions in *Setting Up a List of Values from a Query*.

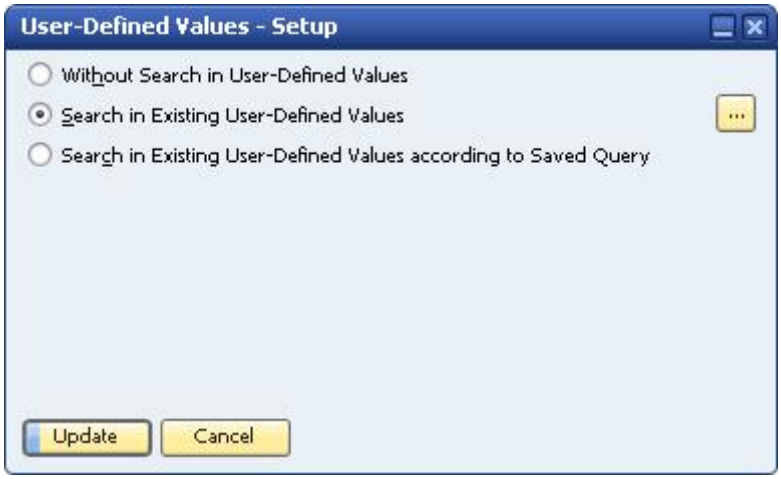
3.1 Setting Up a List of Predefined Values

For a specific field, you can provide a static list of valid values from which the user can choose.

When the user puts the cursor in the field and presses Alt+Shift+F2, a dialog appears from which the user can choose a value.

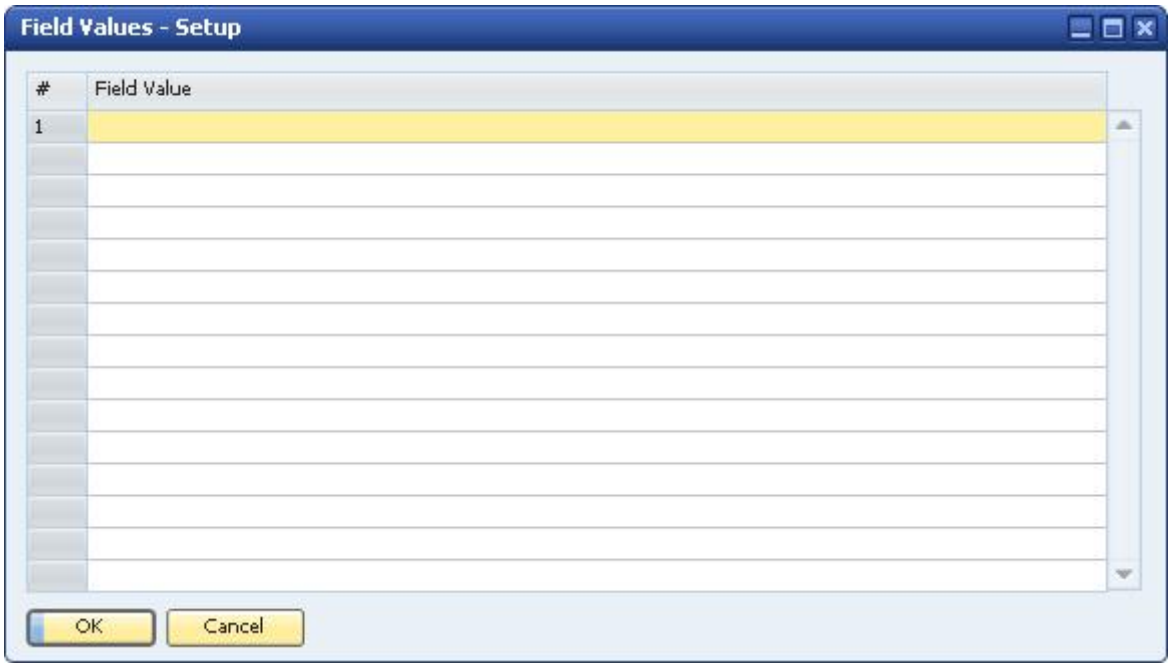
Prerequisite

You have followed the instructions in *Setting Up User-Defined Values*, and selected the *Search in Existing User-Defined Values*.



Procedure

1. Choose . The *Field Values – Setup* window appears.



2. Enter a list of values for the field.

   For each value, press Enter or choose the *Update* button.
3. Choose the *OK* button.
4. Choose the *Update* button.

3.1.1 Adding/Updating Predefined Values

Prerequisites

You have set up a list of predefined values for the field, as described in *Setting Up a List of Predefined Values*.

Procedure

1. Place the cursor in the field for which you want to add/update predefined values, and press Alt+Shift+F2. Alternatively, place the cursor in the field for which you want to add/update predefined values, and choose from the menu bar Tools ® Customization *Tools* ® *User-Defined Values – Setup*.
2. Choose ... . The *Field Values – Setup* window appears, with the cursor in the next available row.

   Note

   You also can open the *Field Values – Setup* window by placing the cursor in the field and pressing Shift+F2. In the *List of User-Defined Values* window, choose the *New* button.

3. Do one of the following:
   - To add a value, enter a value in the next available row, and choose the *Update* button.
   - To update a value, edit the value, and press Enter or choose the *Update* button.
4. Choose the *OK* button.

3.1.2 Deleting Predefined Values

Prerequisites

You have set up a list of predefined values for the field, as described in *Setting Up a List of Predefined Values*.

Procedure

1. Place the cursor in the field for which you want to add/update predefined values, and press Alt+Shift+F2. Alternatively, place the cursor in the field for which you want to add/update predefined values, and choose from the menu bar *Tools* ® *Customization Tools* ® *User-Defined Values – Setup*.
2. Choose ... . The *Field Values – Setup* window appears.
3. Delete a row by right-clicking the row and selecting *Remove*. Alternatively, you can place the cursor in the row, and from the menu bar select *Data* ® *Remove*.
4. Choose the *Update* button.
5. Choose the *OK* button.

3.2 Setting Up a List of Values from a Query

Instead of creating a static list of valid values for a field, you can derive the valid values from a query, which can be run each time the user edits the field. Queries can return one or more values calculated from data in the SAP Business One database or from data entered into fields in the current form but not yet saved to the database.

For a specific field, you can use the values returned by the query in one of the following ways:

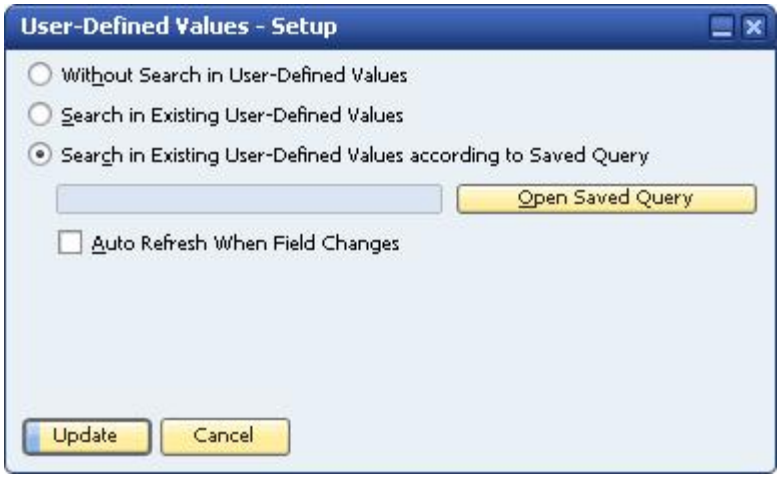
- You can display a list of values derived from a query and enable the user to select a value.
- You can set the field's value automatically whenever a related trigger field is updated. When the user updates the trigger field, the query runs and the returned value is entered into the current field.

  For example, you could set up user-defined values for the *ZIP Code* field within an address in the *Business Partner Master Data* window and make the trigger field the *State* field. When a state is entered, a query runs and retrieves the ZIP code based on the state, and this value is entered into the *ZIP Code* field.

For more information on how to write queries, see *Writing Queries*.

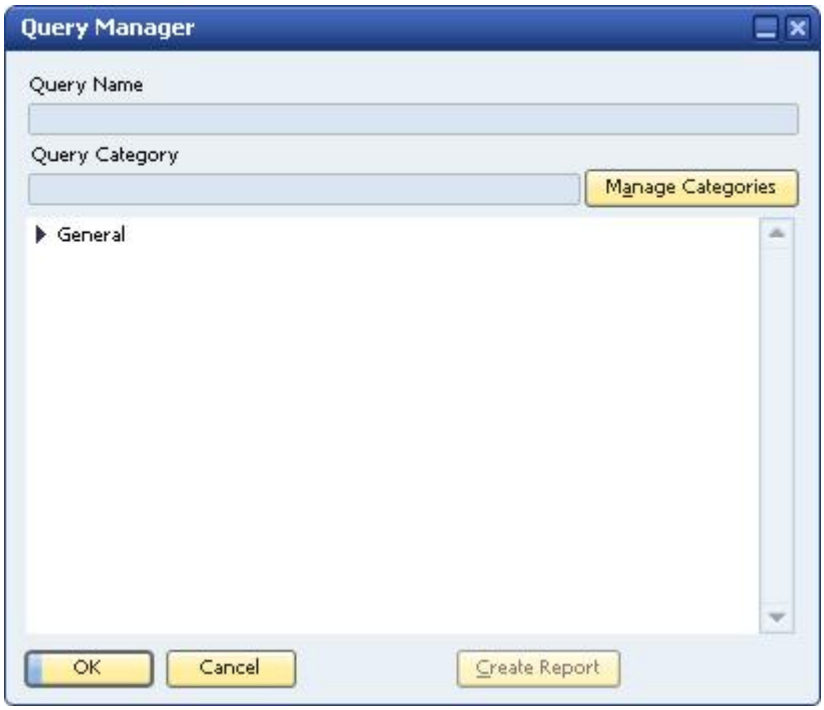
Prerequisite

You have followed the instructions in *Setting Up User-Defined Values*, and selected *Search in Existing User-Defined Values* according to *Saved Query*.



Procedure

1. Choose the *Open Saved Query* button. The *Query Manager* window opens.



2. Select the query from which to derive the list of values for this field and choose the *OK* button.

   The query name is entered in the field in the *User-Defined Values – Setup* window.

   > **Note**
   >
   > To select a different query, choose the *Open Saved Query* button again.

3. Set whether users will select a value from the list generated by a query, or whether the value returned by the query will be entered into the field automatically when another field's value changes.

   Do one of the following:

   - To enable the user to choose from the list of values returned by the query, leave the *Auto Refresh When Field Changes* field unselected.

     The user then must place the cursor in the field, display the list of values by pressing `Shift`+`F2`, and select a value.

   - To automatically update the field with the value in the list, select the *Auto Refresh When Field Changes* field.

     Additional fields are displayed for configuring *Auto Refresh When Field Changes*, as described in *Auto Refresh*.

3.2.1 Auto Refresh

Selecting the *Auto Refresh When Field Changes* checkbox causes the value of the field to be set automatically with the value returned by the query. The user does not need to place the cursor in the field, press `Shift`+`F2` and select a value.

Note

The value is the first field of the first record returned by the query. If you use Auto Refresh, use a query that returns one field and one record.

Note

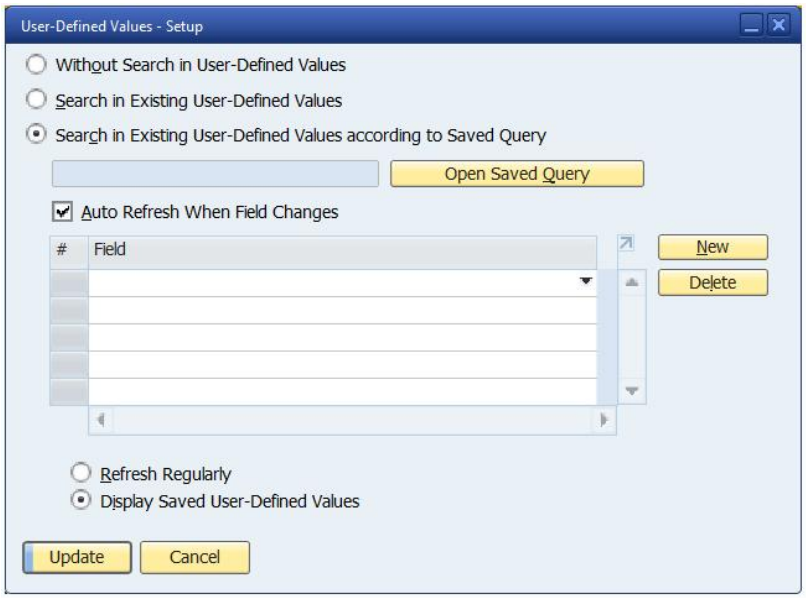
Some system fields are automatically updated when other fields are changed. For example, a change in the *Discount* field in a document row causes a change in the *Price after Discount* field.

Therefore, if you set a user-defined values query for field C to be triggered by a change in field B (for example, *Price after Discount*), field C may also change due to a change in field A (for example, *Discount*).

Procedure

1. Select the field(s) that will act as a trigger for setting the current field's value.

   You can add up to 5 different trigger fields. When any of the trigger fields is updated, the query is run, and the current field's value is set.



Note

Note the following about the drop-down list of a trigger field:

- The list includes only fields that can be updated. You cannot select, for example, *G/L Account Code* in an existing journal entry or *Item Number* in an existing delivery.
- If the current field is a title field, the list includes only title fields.

  If the current field is a row field, the trigger field can be either a title field or a row field.

2. Specify whether the query is run (and the value is updated) whenever a record is displayed, or only when adding a record.

   Select one of the following radio buttons:

   - *Refresh Regularly* - Reruns the query each time you open or browse to a document or display the window.

Note

Be careful when using the *Refresh Regularly* option:

- If you manually change a value set by a user-defined value query and save the document, the next time you display the document, the query is rerun and the value reset, which may not be the intended behavior.
- Each time you browse to a document, all user-defined value queries in the document are activated, possibly causing the document to be displayed slowly.

- *Display Saved User-Defined Values* - Runs the query only when a value is manually changed in the trigger field.

  When browsing documents, the query is not automatically run for existing documents and the form displays the value saved in the field when the document was added or last updated – that is, the value saved in the database.

Auto Refresh + *Refresh Regularly* is similar to *Auto Refresh* + *Display Saved User-Defined Values*, except that *Display Saved User-Defined Values* does not refresh the value when browsing or finding a document.

Example

The following example shows the difference between *Refresh Regularly* and *Display Saved User-Defined Values*. The example adds to the Remarks field the current balance of the business partner for the current sales order.

1. Create the following query:

   ```
   SELECT T0.Balance FROM OCRD T0 WHERE T0.CardCode = $[ORDR.CardCode]
   ```

   The query returns the balance of the business partner for the current document.
2. Create user-defined values for the *Remarks* field based on this query and using *Auto Refresh*.
3. Open the *Sales Order* window and navigate to a document.
4. Place the cursor in the *Remarks* field and press `Shift`+`F2`. The current balance of the business partner is displayed in the field.
5. Add some credits or debits to the customer account in order to change his balance, and then navigate to the same sales order.

   If *Refresh Regularly* was selected when the user-defined values were created, the *Remarks* field is updated with the current account balance, overwriting the value displayed when the sales order was issued. If you navigate to the next record, you will be asked whether this record should be saved.

   If *Display Saved User-Defined Values* was selected, the query is not run and the field is not updated. Instead, the field shows the previous old balance.

3.2.2 Using User-Defined Values

This section describes how to retrieve user-defined values for a field.

Procedure

1. Place the cursor in a field with user-defined values.

2. Press Shift+F2 (or select from the menu bar *Tools* ® *User-Defined Values*).

   If multiple user-defined values are defined without *Auto Refresh*, then the *List of User-Defined Values* window appears, displaying the values.

> **Note**
>
> If only one value is defined, or *Auto Refresh* has been selected, then a value is automatically entered into the field and the window does not appear.

3. Choose a value from the list.

   The value is entered in the field.

4 Writing Queries

Queries look up information from within SAP Business One. Mainly, queries are used to:

- Display information
- Create reports
- Define user-defined values for fields

SAP Business One queries are written in SQL, using one of the following tools:

- Query Wizard: An easy-to-use wizard for creating simple queries, and which requires no knowledge of the SQL language.
- Query Generator: A simple tool that enables the fast creation of more complex SQL statements.

All of the examples in this document use the Query Generator to build queries.

Syntax

The SQL used in SAP Business One enables you to retrieve information from the following sources:

- *SAP Business One Database*

  Database tables that contain stored data about standard SAP Business One objects, such as business partners, employees, invoices, sales orders and purchasing orders.

- *Field in Current Active Form*

  You may want to retrieve data entered into a field but not yet saved to the database.

  For example, you may want to automatically set the *Due Date* field to 20 days after the *Posting Date* field, using a user-defined value query. Therefore, in the query, you would need to retrieve the *Posting Date*.

  The information in a field of a new record is not yet stored in the database, so a standard database query cannot be used to retrieve it.

To build queries, you need to know the following:

- How to write standard SQL.
- How to write special SQL for retrieving field values from SAP Business One forms.

  SAP Business One provides two different ways to specify within a query the value contained in a field on the current form:

  - `$[Tablename.Fieldname]`
    - Tablename: The main table of the active window.
    - Fieldname: The field from the active window.
  - `$[$FieldIndex.FieldColumn.Number/Currency/Date/0]`

    This syntax uses the field's index instead of the specific document table.

    This query syntax includes 3 elements:
    - Field Index: The item number of the field on the form.
    - Field Column: The index of the column of the field within a table.

If the required field is a title field, set the column to 0.

- Number / Currency / Date / 0

  Use Number if the required field includes a number and a currency symbol (item price, document total) to retrieve the number (without the currency symbol or the unit of measurement). This separation is essential when you need to use the number for arithmetic expressions.

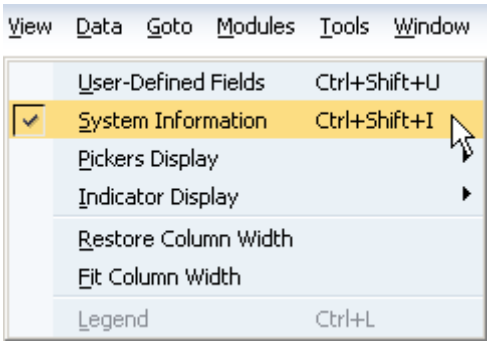
  Use Currency to retrieve the currency symbol or the unit of measurement from a field that includes a number and a currency symbol/unit of measurement.

  Use Date if the required field is a date field and you need to use it for computation.

  Use 0 if the field contains a single value (quantity, item number, item description, business partner code, or document number) that you want to use as a character value.

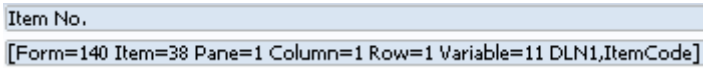
- How to find out to what database table a field is tied and how to reference a field.

  To display information about fields, select from the menu bar *View* ® *System Information*:



The system displays information in the status bar about the item at the location of the cursor.

For example, if you place the cursor over the *Vendor* field in an *A/P Invoice* form, the following displays:



The status bar shows:

- The number of the current form (140)
- The item number of the selected field (38)
- The table that holds the value for this field (DLN1)
- The database field that holds the value for this field (ItemCode)
- The pane in which the field is located (if the field is in a pane)
- The column in which the cursor is located (if the field is in a table)
- The row in which the cursor is located (if the field is in a table)

4.1 Sample Queries

4.1.1 Example 1 – Copy field value to another field

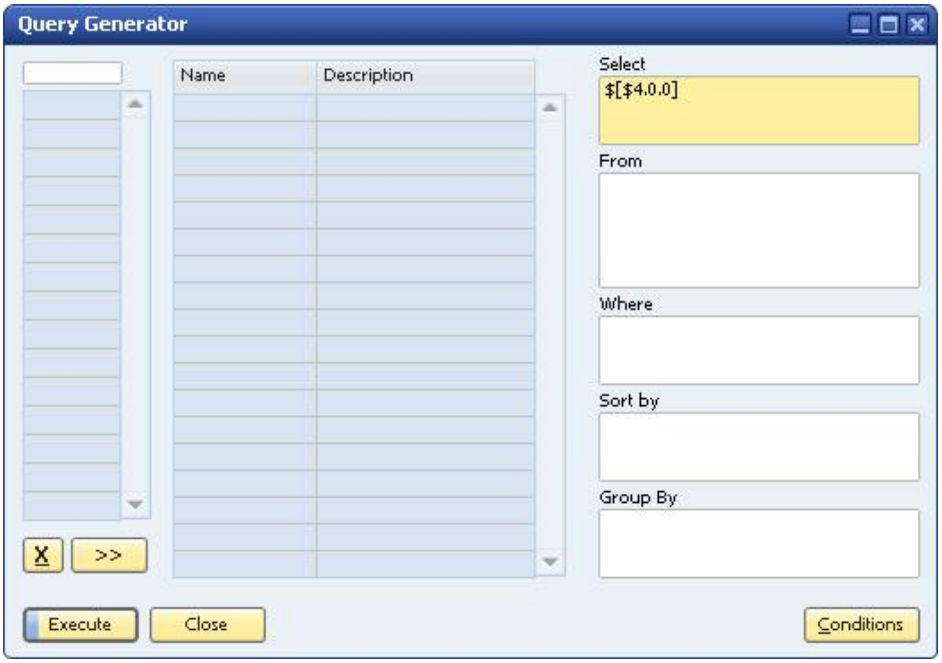
Objective

Retrieve the business partner code from the A/R Invoice form and enter it in the *Remarks* field in the A/R Invoice window.

Procedure

1. Using the Query Generator, create and save a query with the following in the *Select* area:

   ```
   $[$4.0.0] or $[OINV.CARDCODE]
   ```



2. Open the *A/R Invoice* window.
3. Enter a business partner code in the *Customer* field.
4. Place the cursor in the *Remarks* field, press Alt+Shift+F2, and do the following:
   1. Select *Search in Existing User-Defined Values according to Saved Query*.
   2. Select the saved query.
   3. Select *Auto Refresh When Field Changes*.
   4. Click the dropdown list and select *Customer/Vendor Code*.
   5. Select *Display Saved User-Defined Values*.
   6. Choose *Update*.

To test, change the customer code in the invoice. Changing the customer code updates the *Remarks* field automatically.

> **Recommendation**
>
> Use the format `$[$4.0.0]` rather than `$[OINV.CARDCODE]` because the first can be linked to search fields in any sales or purchasing document while the second can be linked only to A/R invoices.

4.1.2 Example 2 – Copy field value to user-defined field

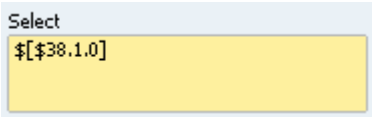
Objective

Retrieve the item number from an active Sales Order window and enter it in a user-defined field.

Procedure

1. Create a user-defined field in *Marketing Documents - Rows*. Since the user-defined field properties need to be identical to the Item Number field properties, set the following for the field:
   - Type: Alphanumeric
   - Structure: Regular
   - Length: 20
2. Using the Query Generator, create and save a query with the following in the *Select* area:

   `$[$38.1.0] or $[RDR1.ITEMCODE]`

   The number 38 represents for the item table, and 1 represents for the *Item No.* column.
3. Open the *Sales Order* window:
   - Type a customer code in the *Customer* field.
   - Type an item number in the *Item No.* column.
   - Place the cursor in the user-defined field, press `Alt`+`Shift`+`F2`, and do the following:
     1. Select *Search in Existing User-Defined Values according to Saved Query*.
     2. Select the saved query.
     3. Select *Auto Refresh When Field Changes*.
     4. Select *When Exiting Altered Column*.
     5. Select the field *Item No.* field.
     6. Select *Display Saved User-Defined Values*.
     7. Choose *Update*.

To test, change the item number. Changing the item number updates the search field automatically.

4.1.3 Example 3 – Copy row field value to user-defined field

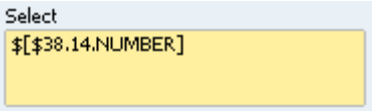
Objective

Retrieve the unit price (the numeric value) from a sales or purchasing document window and enter it in a user-defined field.

Procedure

1. Create a user-defined field in *Marketing Documents - Rows* with the following settings:
   - Type: Alphanumeric
   - Structure: Regular
2. Using the Query Generator, create and save a query with the following in the *Select* area:

   `$[$38.14.NUMBER]`

   The number 38 represents for the item table, 14 represents for the *Unit Price* column, and `NUMBER` indicates the price's numeric value.
3. Open any sales or purchasing document window.
4. Type a business partner code in the *Customer* or *Vendor* field.
5. Type an item number in the *Item No.* column.
6. Place the cursor in the user-defined field, press `Alt`+`Shift`+`F2`, and do the following:
   1. Select *Search by Saved Query*.
   2. Select the saved query.
   3. Select *Auto Refresh*.
   4. Select *When Exiting Altered Column*.
   5. Select the field *Unit Price* field.
   6. Select *Display Saved User-Defined Values*.
   7. Choose *Update*.

To test, change the price. Changing the price updates the user-defined field automatically.

> **Note**
>
> The numeric value retrieved from the *Unit Price* field can be used for arithmetic expressions.

4.1.4 Example 4 – Copy currency symbol

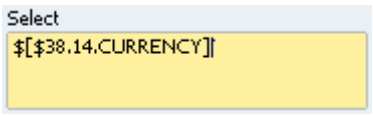
Objective

Retrieve the unit price currency (the currency symbol) from a sales or purchasing document window and enter it in a user-defined field.

Procedure

1. Create a user-defined field in *Marketing Documents - Rows* with the following settings:
   - Type: Alphanumeric
   - Structure: Regular
   - Length: 3
2. Using the Query Generator, create and save a query with the following in the *Select* area:

   ```
   $[$38.14.CURRENCY]
   ```



3. Open any sales or purchasing document window.
4. Type a business partner code in the *Customer* or *Vendor* field.
5. Type an item number in the *Item No.* column.
6. Place the cursor in the user-defined field, press `Alt`+`Shift`+`F2`, and do the following:
   1. Select *Search by Saved Query*.
   2. Select the saved query.
   3. Select *Auto Refresh*.
   4. Select *When Exiting Altered Column*.
   5. Select the field *Unit Price* field.
   6. Select *Display Saved Values*.
   7. Choose *Update*.

To test, change the price currency. Changing the price currency will update the user-defined field automatically.

4.1.5 Example 5 – Copy a user-defined field's value to a system field

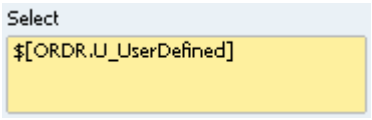
Objective

Retrieve a user-defined field's value from an active *Sales Order* window and enter it in the *Remarks* field of the *Sales Order* window.

Procedure

1. Create a user-defined field called `UserDefined` in the *Marketing Documents – Title* category. For more information, see the how-to guide *How to Create User-Defined Fields and Tables*.
2. Using the Query Generator, create and save a query with the following in the *Select* area:

   `$[ORDR.U_UserDefined]`

   Note

   To specify within a query the value contained in a user-defined field on the current form, you must use the `$[Tablename.Fieldname]` format of the query.
3. Open the *Sales Order* window:
   1. Type a customer code in the *Customer* field.
   2. Type an item number in the *Item No.* column.
   3. Type a value in the *UserDefined* field.
   4. Place the cursor in the *Remarks* field, press `Alt`+`Shift`+`F2`, and do the following:
      1. Select *Search in Existing User-Defined Values* according to *Saved Query*.
      2. Select the saved query.
      3. Select *Auto Refresh When Field Changes*.
      4. Select *When Exiting Altered Column*.
      5. Select the *UserDefined* field.
      6. Select *Display Saved User-Defined Values*.
      7. Choose *Update*.

To test, change the value in the *UserDefined* field. Changing the value should update the *Remarks* field automatically.

4.1.6 Example 6 – Calculate value from field value

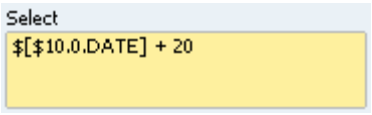
Objective

In a sales or purchasing document, retrieve the value in the *Posting Date* field and enter the `Posting Date + 20` into the *Due Date* field.

Procedure

1. Using the Query Generator, create and save a query with the following in the *Select* area:

   `$[$10.0.DATE] + 20`



2. Open a sales or purchasing document window.
3. Type a business partner code in the *Customer* or *Vendor* field.
4. Place the cursor in the *Due Date or Delivery Date* field (depending on the document type), press `Alt`+`Shift`+`F2`, and do the following:
   1. Select *Search in Existing User-Defined Values according to Saved Query*.
   2. Select the saved query.
   3. Select *Auto Refresh When Field Changes*.
   4. Select the field *Posting Date* field.
   5. Select *Display Saved User-Defined Values*.
   6. Choose *Update*.

To test, change the posting date, which updates the *Due Date or Delivery Date* field.

4.1.7 Example 7 – Calculate numerical value from field value

Objective

Retrieve the tax amount per 1 item unit from a sales or purchasing documents and enter it in a user-defined field.

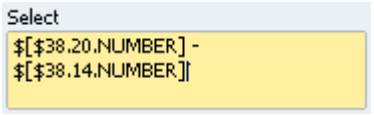
> **Note**
>
> To get the tax amount per 1 item unit, the price before tax has to be subtracted from the gross price.

Procedure

1. Create a user-defined field in *Marketing Documents - Rows* with the following settings:
   - Type: Units and Totals
   - Structure: Amount
2. Using the Query Generator, create and save a query with the following in the *Select* area:

   ```
   SELECT $[$38.20.NUMBER] - $[$38.14.NUMBER]
   ```

   > **Note**
   >
   > For calculated fields, you must place the word SELECT before the calculation clause, and inside the brackets the expression must end with .NUMBER.

   The string $[$38.14.NUMBER] represents the price before tax.

   The string $[$38.20.NUMBER] represents the gross price.
3. Open a sales or purchasing document window.
4. Type a business partner code in the *Customer* or *Vendor* field.
5. Type an item number in the *Item No.* column.
6. Place the cursor in the user-defined field, press Alt+Shift+F2, and do the following:
   1. Select *Search by Saved Query*.
   2. Select the saved query.
   3. Choose *Update*.

   > **Recommendation**
   >
   > When you are calculating two fields, auto refresh is not recommended because the user-defined field may not update correctly. For example, using the query in the example above changes the tax field but not the price field, because the trigger is the price field. In such cases, we recommend updating the field manually, either by pressing Shift+F2 and not using auto refresh or by making sure that the user-defined field was updated correctly.

To test, change the item price. Changing the item price updates the user-defined field.

4.1.8 Example 8 – Cast numerical value

Objective

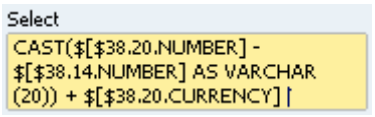
Add the currency symbol to the value retrieved in Example 6.

Since the tax amount is a numeric value and the currency symbol is an alphanumeric value, you need to use the SQL function CAST.

Procedure

1. Create a user-defined field in *Marketing Documents - Rows* with the following settings:
   - Type: Alphanumeric
   - Structure: Regular
2. Using the Query Generator, create and save a query with the following in the *Select* area:

   ```
   SELECT CAST($[$38.20.NUMBER] - $[$38.14.NUMBER] AS VARCHAR(20)) +
   $[$38.20.CURRENCY]
   ```

   Note

   `AS VARCHAR (20)` indicates the size allocated to the subtraction result.
3. Open a sales or purchasing document window.
4. Type a business partner code in the *Customer* or *Vendor* field.
5. Type an item number in the *Item No.* column.
6. Place the cursor in the user-defined field, press `Alt`+`Shift`+`F2`, and do the following:
   1. Select *Search by Saved Query*.
   2. Select the saved query.
   3. Choose *Update*.

4.1.9 Example 9 – Look up value from database

Objective

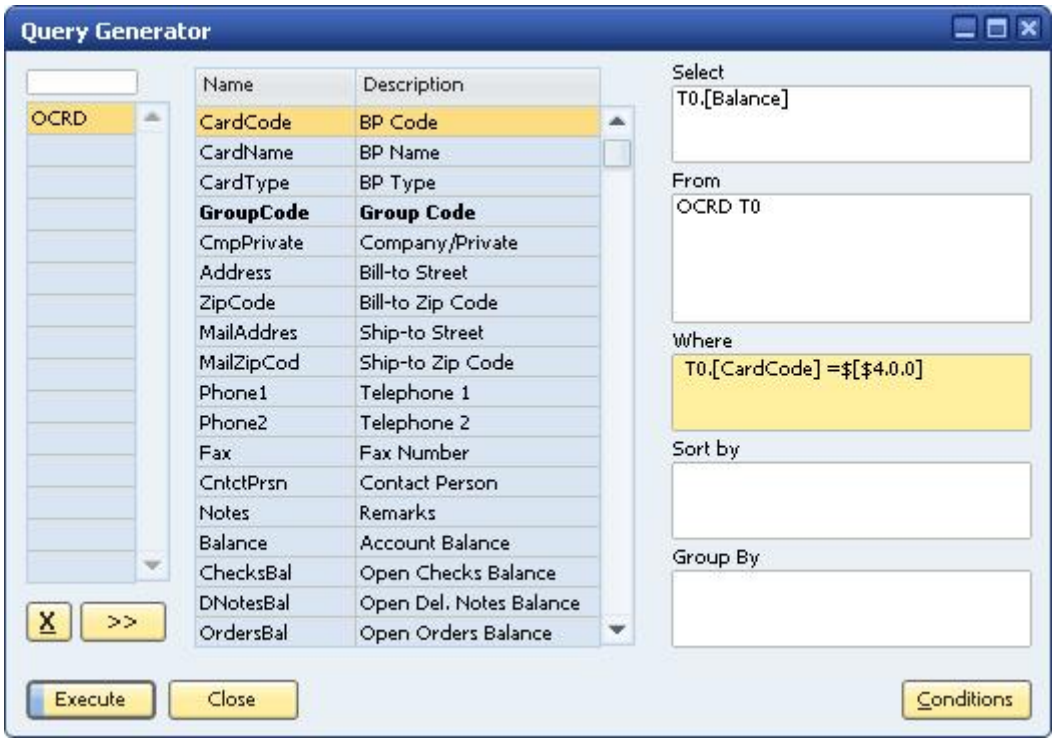
Retrieve the business partner's account balance and enter it in a sales or purchasing document.

Procedure

1. Create a user-defined field in the *Marketing Documents – Title* category with the following settings:
   - Type: Units and Totals
   - Structure: Amount

   For more information, see the how-to guide *How to Create User-Defined Fields and Tables*.
2. Using the Query Generator, create and save the following query:

   ```
   SELECT T0.[Balance] FROM OCRD T0 WHERE T0.[CardCode] = $[4.0.0]
   ```



3. Create user-defined values for the user-defined field using the query above, and select one of the following *Auto Refresh options*:
   - No Auto Refresh.

     The current business partner balance is entered into the field each time you click the user-defined field and press `Shift`+`F2`.
   - *Auto Refresh and Refresh Regularly*

     The user-defined field is refreshed each time the trigger field is updated.

     When creating the record, the current balance is entered, and each time you browse or find the document, the user-defined field is updated with the latest balance.
   - *Auto Refresh and Display Saved User-Defined Values*

     The user-defined field is refreshed each time the trigger field is updated.

     The current business partner balance is entered into the field. However, the balance is not refreshed each time you browse or display the document. You can update the field manually.

4.1.10 Example 10 – Create list of values from database lookup

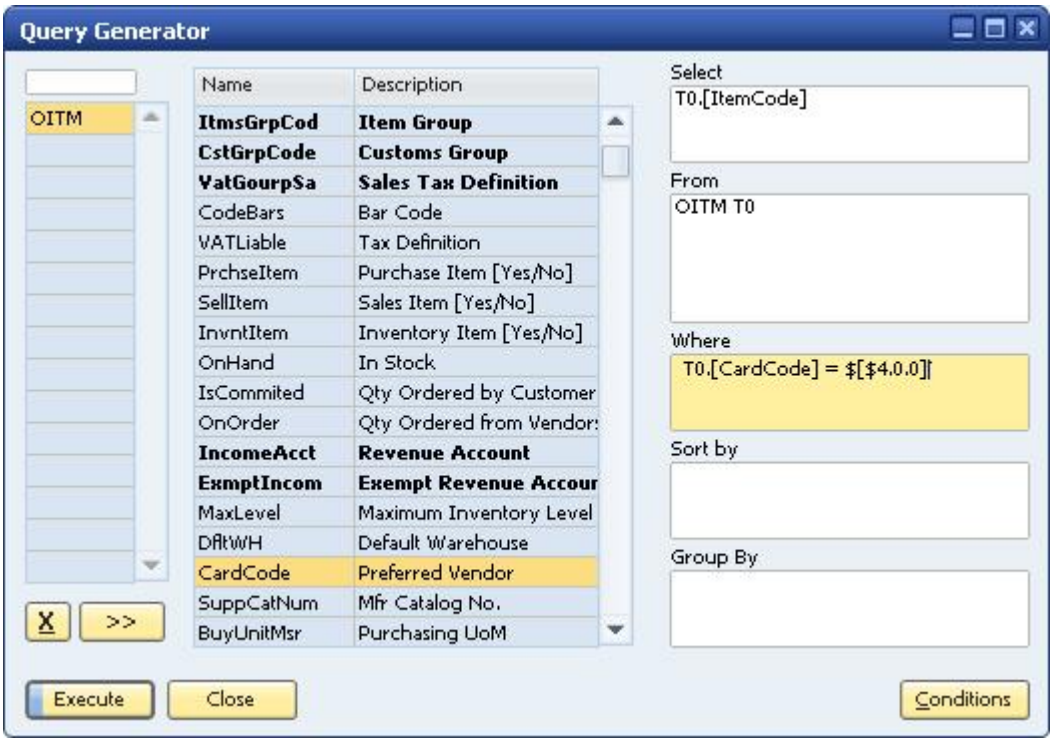
Objective

Provide the user with a list of items to which the *Preferred Vendor* is identical to the *Vendor Code* selected in a purchasing document.

Procedure

1. Using the Query Generator, create and save the following query:

   ```
   SELECT T0.[ItemCode] FROM OITM T0 WHERE T0.[CardCode] = $[$4.0.0]
   ```



2. Open a purchasing document window.
3. Type a business partner code in the *Vendor* field.
4. Place the cursor in the *Item No.* field of the table in the *Contents* tab, press `Alt`+`Shift`+`F2`, and do the following:
   1. Select *Search by Saved Query*.
   2. Select the saved query.
   3. Choose the *Update* button.

To test, place the cursor in the *Item No.* field and press `Shift`+`F2`. A list of all the items for which the current vendor is the preferred vendor is displayed.

4.1.11 Example 11 – Create query with user-defined table

Objective

Select the ZIP code of a business partner from predefined values stored in the database based on the business partner's country, city and address.

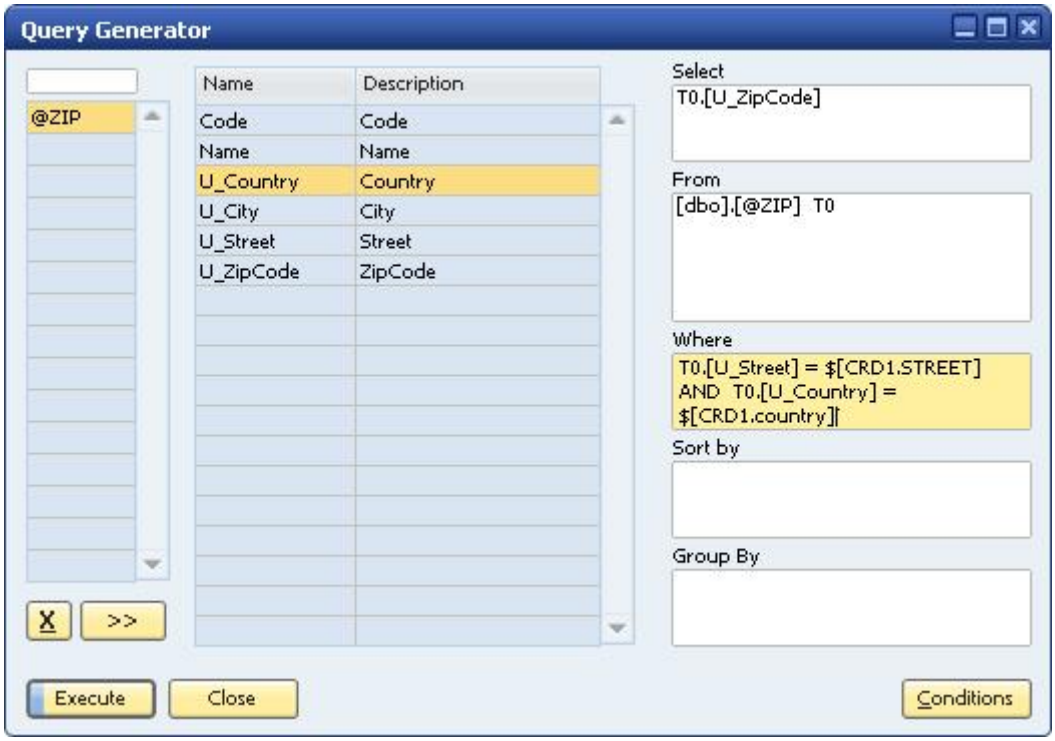
Procedure

1. Create a user-defined table called *Zip* and add to it the user-defined fields *Country*, *City*, *Street*, and *ZipCode*.
2. Add data to the table manually (*Tools* ® *User-Defined Windows*) or automatically (by importing data into the `@ZIP` table using an existing list of addresses).

   > **Note**
   >
   > In the *Country* field, enter the country code as defined in *Administration* ® *Setup* ® *Business Partners* ® *Countries* ® *Countries – Setup*.

3. Using the Query Generator, create and save a query with the following in the *Select* area:

   ```
   SELECT T0.[U_ZipCode] FROM [dbo].[@ZIP]  T0 WHERE T0.U_City = $[CRD1.City] AND
   T0.U_STREET = $[CRD1.STREET] and T0.U_Country = $[CRD1.country]
   ```



4. Open the *Business Partner Master Data* window.
5. Switch to *Add* mode.
6. Choose the *Addresses* tab.
7. Enter a name in the *Address Name* field.
8. Enter values in the *Street*, *City* and *Country* fields.
9. Place the cursor in the *Zip Code* field, press `Alt`+`Shift`+`F2`, and do the following:
   1. Select *Search by Saved Query*.
   2. Select the saved query.
   3. Choose the *Update* button.
10. Place the cursor in the *Zip Code* field and press `Shift`+`F2`.

    A zip code is retrieved from the user table `@ZIP`.

4.1.12 Example 12 – Retrieve current user

Objective

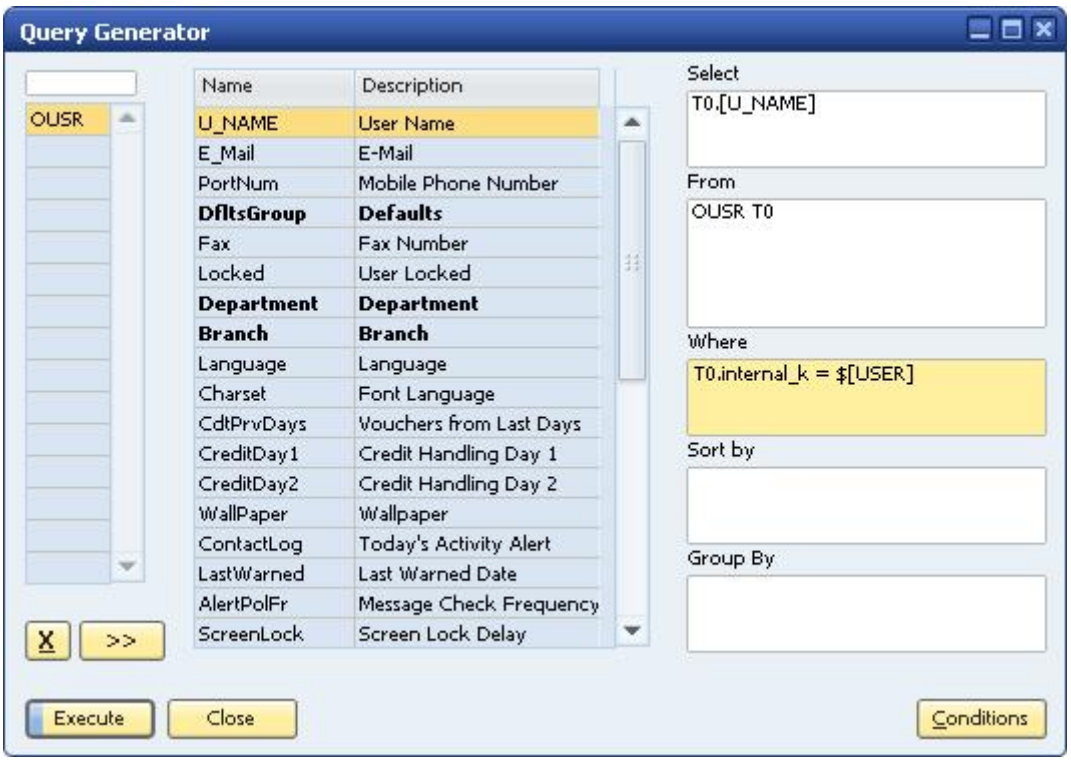
For new documents, display the document creator's name in the *Remarks* field.

> **Note**
>
> - You can retrieve the name of the current user logged into SAP Business One by using the string `$[USER]`.
> - The field `internal_k` is the user internal number in the `OUSR` table. This is the key of the table. To save information about the user who performed the operation anywhere in the database, use the `internal_k` field.

Procedure

1. Using the Query Generator, create and save the following query:

   ```
   SELECT T0.u_name FROM OUSR T0 WHERE T0.internal_k = $[USER]
   ```



2. Open a sales or purchasing document.
3. Place the cursor in the *Remarks* field, press `Alt`+`Shift`+`F2`, and do the following:
   1. Select *Search by Saved Query*.
   2. Select the saved query.

3. Select *Auto Refresh When Field Changes*.
4. Click the drop-down list and select *Customer/Vendor Code*.
5. Select *Display Saved User-Defined Values*.
6. Choose the *Update* button.

4.1.13 Example 13 – Set field value to complex string

Objective

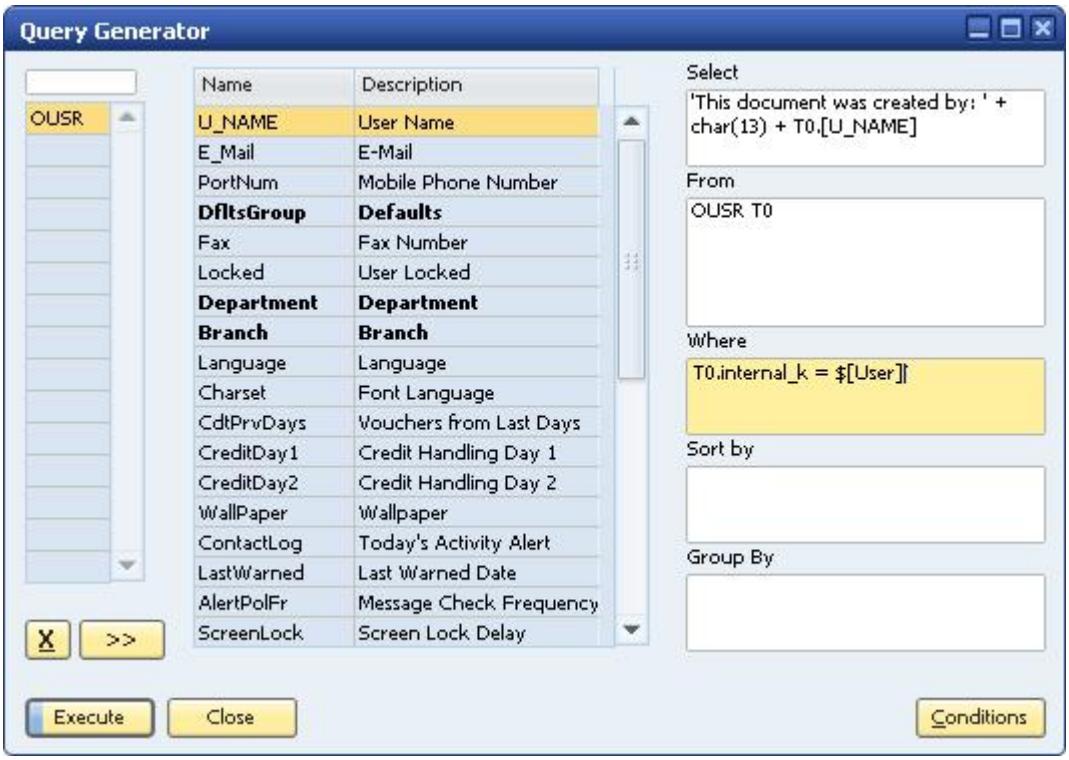
Display the user name in a document along with a text string and proper breaks between the text and the user name.

The value to be entered is: `This document was created by: <User Name>`

Procedure

Following the instructions for Example 13, but use the following query:

```
SELECT 'This document was created by: ' + char(13) + T0.[U_NAME] FROM OUSR T0 WHERE T0.internal_k = $[User]
```



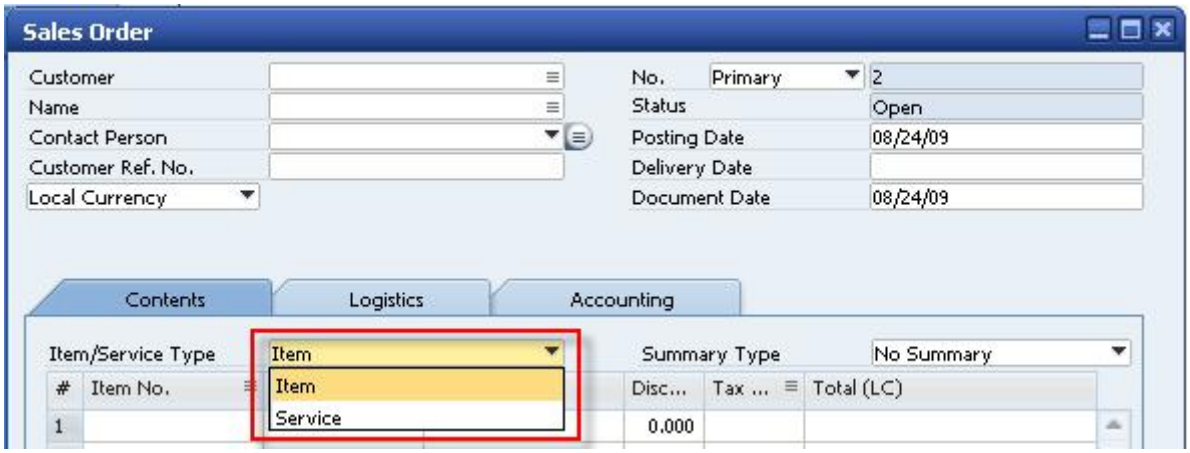
4.2 Field Indexes

This section lists the item IDs of commonly used fields in marketing documents.

The IDs can be used in queries that include references to fields in the current form, as described in *Writing Queries*. For example, `$[$54.0.0]` retrieves the value entered in the *Name* field in a document.

The list is divided into the following sections:

- Item Type Documents: Sales and purchasing documents where the *Item/Service Type* field is set to **`Item`**.
- Service Type Documents: Sales and purchasing documents where the *Item/Service Type* field is set to **`Service`**.



Title fields are listed with a value for the field index. You set the column index to 0 for title fields. Row fields are listed with values for both the field and column indexes.

4.2.1 Item Type Documents

Field Name	Field Index	Column Index
Items	3	
Summary Type	75	
Currency Type	70	
Document Currency	63	
Customer/Vendor (text)	5	
Customer/Vendor	4	
Name (text)	55	
Name	54	
Contact person (text)	83	
Contact person	85	

Field Name	Field Index	Column Index
Manual	78	
No. (text)	9	
No.	8	
Posting Date (text)	11	
Posting Date	10	
Delivery Date (text)	13	
Delivery Date	12	
Customer Ref. No. (text)	15	
Customer Ref. No.	14	
Series (text)	84	
Series	88	
Code/Catalog no.	79	
Item Number	38	1
Catalog No.	38	2
Item Description	38	3
Bar Code	38	4
Catalog No.	38	5
Serial Number	38	6
Factor 1	38	7
Factor 2	38	8
Factor 3	38	9
Factor 4	38	10
Quantity	38	11
Base Units	38	12
No. of Packages	38	13
Price w/o Dscnt	38	14
Discount %	38	15
Rate	38	16
Price	38	17
Tax Code	38	18
Price Including Tax	38	20

Field Name	Field Index	Column Index
Total (LC)	38	21
Total (SC)	38	22
Total (Doc)	38	23
Whse	38	24
Del. Date	38	25
SE Code	38	26
SE	38	27
Comm. %	38	28
Acct. No.	38	29
Costing Code	38	30
Project Code	38	31
Open Qty.	38	32
On Hand	38	33
Sales Order	38	34
Committed	38	35
COGM	38	37
BOM Type	38	39
Tgt. Tab.	38	41
Target Key	38	42
Base Type	38	43
Base Ref.	38	44
Base Key	38	45
Base Row	38	46
BP Card	38	49
Length	38	53
Width	38	54
Height	38	55
Volume	38	56
Vol. Unit.	38	57
Weight	38	58
Total Tax (LC)	38	82

Field Name	Field Index	Column Index
Total Tax (Doc.)	38	83
Block No.	38	86
Import Log	38	87
Acquis.	38	88
Total Before Discount (text)	23	
Total Before Discount	22	
%Discount (text)	25	
%Discount	24	
Discount Sum	42	
Tax (text)	99	
Tax	27	
Total (text)	30	
Total	29	
Paid/Credited (text)	32	
Paid/Credited	31	
Balance Due (text)	34	
Balance Due	33	
Remarks (text)	17	
Remarks	16	
Journal Remark (text)	19	
Journal Remark	18	
Sales Employee (text)	21	
Sales Employee	20	
Payment Terms (text)	48	

4.2.2 Service Type Documents

Field Name	Field Index	Column Index
Items	3	
Summary Type	75	
Currency Type	70	

Field Name	Field Index	Column Index
Document Currency	63	
Customer/Vendor (text)	5	
Customer/Vendor	4	
Name (text)	55	
Name	54	
Contact Emp. (text)	83	
Contact Emp.	85	
Manual	78	
No. (text)	9	
No.	8	
Posting Date (text)	11	
Posting Date	10	
Due Date (text)	13	
Due Date	12	
Customer Ref. No. (text)	15	
Customer Ref. No.	14	
Series (text)	84	
Series	88	
Code/Catalog no.	79	
Description	39	1
G/L Account	39	2
Costing Code	39	3
Project Code	39	4
Price w/o Dscnt	39	5
Discount %	39	6
Rate	39	7
Price	39	8
Tax Code	39	9
Tax%	39	10
Price Including Tax	39	11
Total (LC)	39	12

Field Name	Field Index	Column Index
Total (SC)	39	13
Total (Doc)	39	14
SE Code	39	15
SE	39	16
Commission %	39	17
Open Amount	39	18
Tgt Type	39	21
Target Key	39	22
Base Type	39	23
Base Ref.	39	24
Base Key	39	25
Base Row	39	26
Base Card	39	28
Tax Total (LC)	39	32
Acquis.	39	35
Total Before Discount (text)	23	
Total Before Discount	22	
%Discount (text)	25	
%Discount	24	
Discount Sum	42	
Tax (text)	99	
Tax	27	
Total (text)	30	
Total	29	
Paid/Credited (text)	32	
Paid/Credited	31	
Balance Due (text)	34	
Balance Due	33	
Remarks (text)	17	
Remarks	16	
Journal Remark (text)	19	

Field Name	Field Index	Column Index
Journal Remark	18	
Sales Employee (text)	21	
Sales Employee	20	
Payment Terms (text)	48	
Payment Terms	47	

4.3 Troubleshooting Queries

Before attaching a query to a user-defined value, check that the user-defined query yields the correct results. To do this, place your cursor in the field, from the menu bar choose *Tools* ® *Queries* ® *User Queries*, and select your query. In the open query window, you should see the result of the query. Otherwise, SAP Business One may display an error message in the status bar or simply not produce a result in the field.

> **Note**
>
> Do not make any changes to the query in this window, as the changes would affect only this specific query and not the general query.

This section describes common errors when building queries:

- Converting Data Types:

  An error occurs when attempting to use values with the wrong data type, for example, to perform arithmetic on string values.

  For example, the value `Select $[RDR1.Quantity] * 2` causes an error because you cannot multiply the string `$[RDR1.Quantity]` by the number 2. The correct syntax is:

  `Select $[RDR1.Quantity.Number] * 2`.

- Error Message: *Character number is greater than allowed*

  An error occurs when the result of a user-defined values query is larger than allowed for the field.

  Do one of the following:

  - For user-defined fields, increase the field's length setting.
  - For system fields, modify the query to reduce the length of the result.

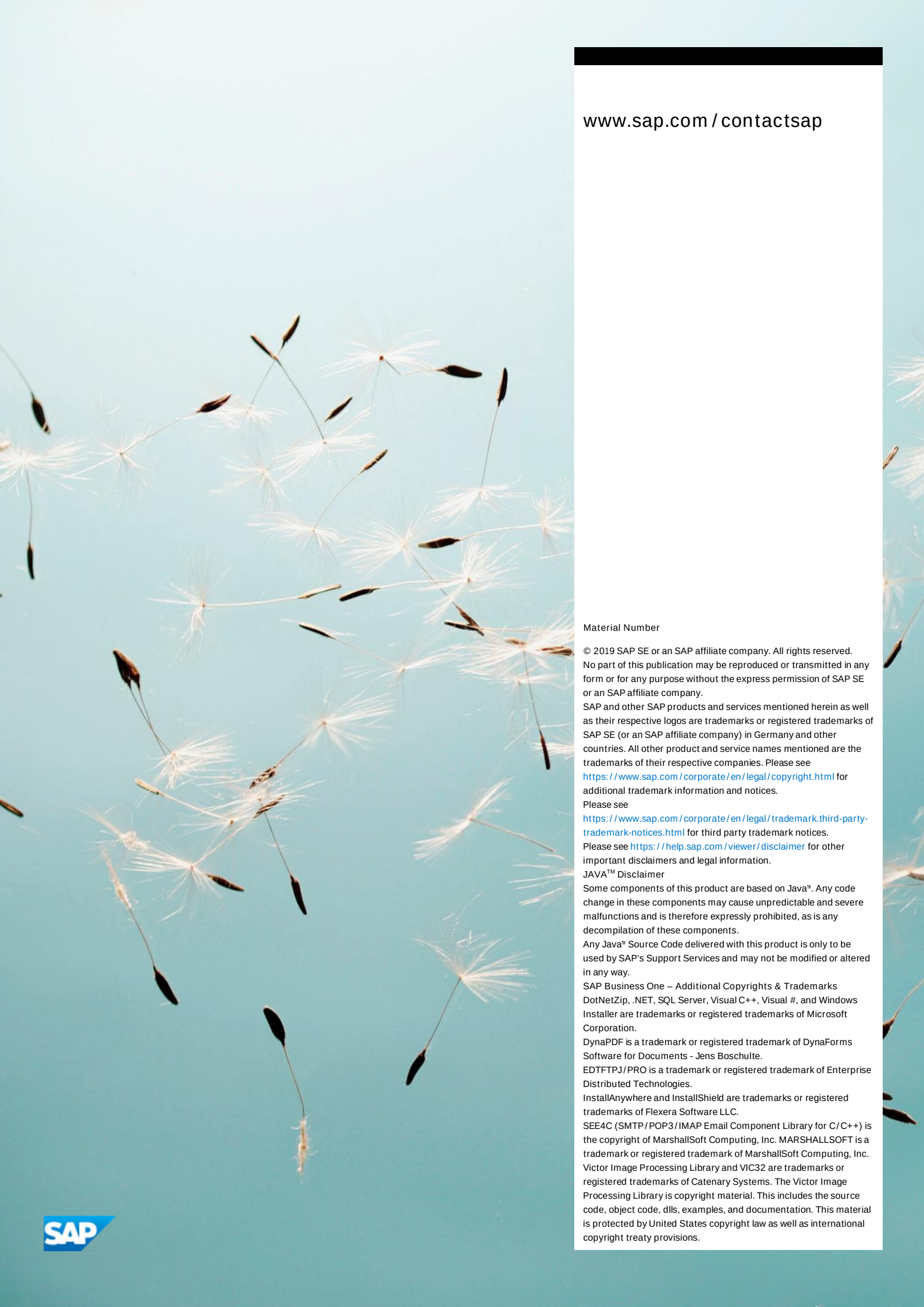
- Error Message: *Statement could not be prepared*

  This error can have several causes. The following are common causes:

  - The query refers to a field that does not exist.

    One reason the field may not exist is that the name of a user-defined field was entered without the `U_` prefix.

  - There is an SQL syntax error.



www.sap.com/contactsap

Material Number