Creatio Academy

Word reports

MS Word reports

Version 8.0

Table of Contents

MS Word reports

Easy

Users can create MS Word reports in Creatio and configure them using the Creatio MS Word Report Designer plug-in. Learn more about creating and setting up MS Word reports in the "Set up MS Word reports" article.

Use macros to set up extra data output options for an MS Word report. You can use basic macros and create custom macros.

Basic macros

The general procedure of creating an MS Word report using basic macros is as follows:

1. Install the MS Word Report Designer plug-in. This is a one-time procedure. Learn more in the "Installing Creatio plug-in for MS Word" article.
2. Add a new report record in the [*Report setup*] section.
3. Set up the report display parameters
4. Set up the report data fields and tables. Add a tag with the name of the macro in the `[#MacrosName#]` format to the column when setting up column fields.
5. Set up the report template layout in the Creatio MS Word Report Designer plug-in and upload the template to Creatio.

The structure of macros in MS Word reports

```
ColumnName[#Macro name|Arguments#]
```

The [#Date#] macro

Converts a date to a specified date format. The default date format is "`dd.MM.yyyy`". If the date format is not specified, the entered date value will be converted to the default format. A detailed description of date formats is available in the Microsoft documentation. The argument is optional.

Examples:

```
ColumnName[#Date#]
```

If the entered value is "`07/15/2020 11:48:24 AM`", the macro will return "`15.07.2020`".

```
ColumnName[#Date|MM/dd/yyyy#]
```

If the entered value is "`31/01/2019 08:25:48 AM`", the macro will return "`01/31/2019`".

The [#Lower#] macro

Converts the value of a string to lowercase. The macro has no arguments.

An example:

```
ColumnName[#Lower#]
```

If the entered value is "`ExaMpLe`", the macro will return "`example`".

The [#Upper#] macro

Converts the value of a string to uppercase. The argument is optional. If the "`FirstChar`" argument is passed to the macro, only the first character in the string will be converted to uppercase.

Examples:

```
ColumnName[#Upper#]
```

If the entered value is "`example`", the macro will return "`EXAMPLE`".

```
ColumnName[#Upper|FirstChar#]
```

If the entered value is "`example`", the macro will return "`Example`".

The [#NumberDigit#] macro

Convert a raw number to a number with digit group separators. The default delimiter is the space character. The argument is optional.

Examples:

```
ColumnName[#NumberDigit#]
```

If the entered value is "`345566777888.567`", the macro will return "`345 566 777 888.567`".

```
ColumnName[#NumberDigit|,#]
```

If the entered value is "`345566777888.567`", the macro will return "`345,566,777,888.567`".

Note. If the fractional part of the number equals zero, only the integer part will be returned. If the entered value is "`345566777888.000`", the macro will return "`345,566,777,888`".

The `[#Boolean#]` macro

Converts a boolean value to a custom representation. The argument is required. The following arguments are available:

- `CheckBox` – converts the entered value to "`☑`" or "`☐`".
- `Yes,No` – converts the entered value to "`Yes`" or "`No`".

Examples:

```
ColumnName[#Boolean|CheckBox#]
```

If the column contains the `true` value, the macro will return "`☑`".

```
ColumnName[#Boolean|Yes,No#]
```

If the column contains the `true` value, the macro will return "`Yes`".

Custom macros

The [*Report setup*] section enables users to create MS Word reports using Creatio tools and configure them using the Creatio MS Word Report Designer plug-in. Learn more about creating and setting up MS Word reports in the "Set up MS Word reports" article.

The general procedure of creating an MS Word report using custom macros is as follows:

1. Install the MS Word Report Designer plug-in. This is a one-time procedure. Learn more in the "Installing Creatio plug-in for MS Word" article.
2. Add a new report record in the [*Report setup*] section.
3. Set up the report display parameters
4. Implement custom macros.
5. Set up the report data fields and tables.
6. Set up the report template layout in the Creatio MS Word Report Designer plug-in and upload the template to Creatio.

A macro for setting up an MS Word report is a class implementing the `IExpressionConverter` interface (see the `ExpressionConverterHelper` schema of the `NUI` package).

To make a custom macro callable from the report template, mark the macro with the `ExpressionConverterAttribute` attribute containing the name of the macro. For example:

```
[ExpressionConverterAttribute("CurrentUser")]
```

The `Evaluate(object value, string arguments = "")` interface method must be implemented in the class. The method accepts an MS Word report template field value as an argument and returns the `string` type value that will be inserted instead of this field in the ready MS Word report.

The general procedure of creating a custom MS Word report macro is as follows:

1. Create a new report in the [*Report setup*] section.
2. Set up the report display parameters.
3. Set up the report data fields and tables.
4. Add the [*Id*] column to the list of report columns that will be the incoming parameter for the custom macro.
5. Add a schema of the [*Source Code*] type with a class implementing the `IExpressionConverter` interface to the custom package. The class must be marked by the `ExpressionConverterAttribute` attribute with the name of the macro. Implement the `Evaluate(object value, string arguments = "")` method in it.
6. Publish the [*Source Code*] type object schema.
7. Add a tag with the name of the custom macro in the `[#MacrosName#]` format to the [*Id*] column when setting up column fields.

Transferring the package to another development environment

To transfer the package with the report to another development environment, go to the [*Configuration*] section -> the [*Data*] tab and bind the data of the following elements:

- `SysModuleReport_ReportName` – the report. To bind it, use the report Id from the [*dbo.SysModuleReport*] database table.
- `SysModuleReportTable_ReportName` – the tabular component of the report. To bind it, use the report Id from the [*dbo.SysModuleReportTable*] database table.

Note. You can view the record Id in the database table even if you do not have access to the database. To do this, display the Id system column in the window of binding data to packages.

Create MS Word report using basic macros

Easy

Case. Create an "Account Info" report for the [*Accounts*] section edit page to display the following information about the account:

- [*Name*].

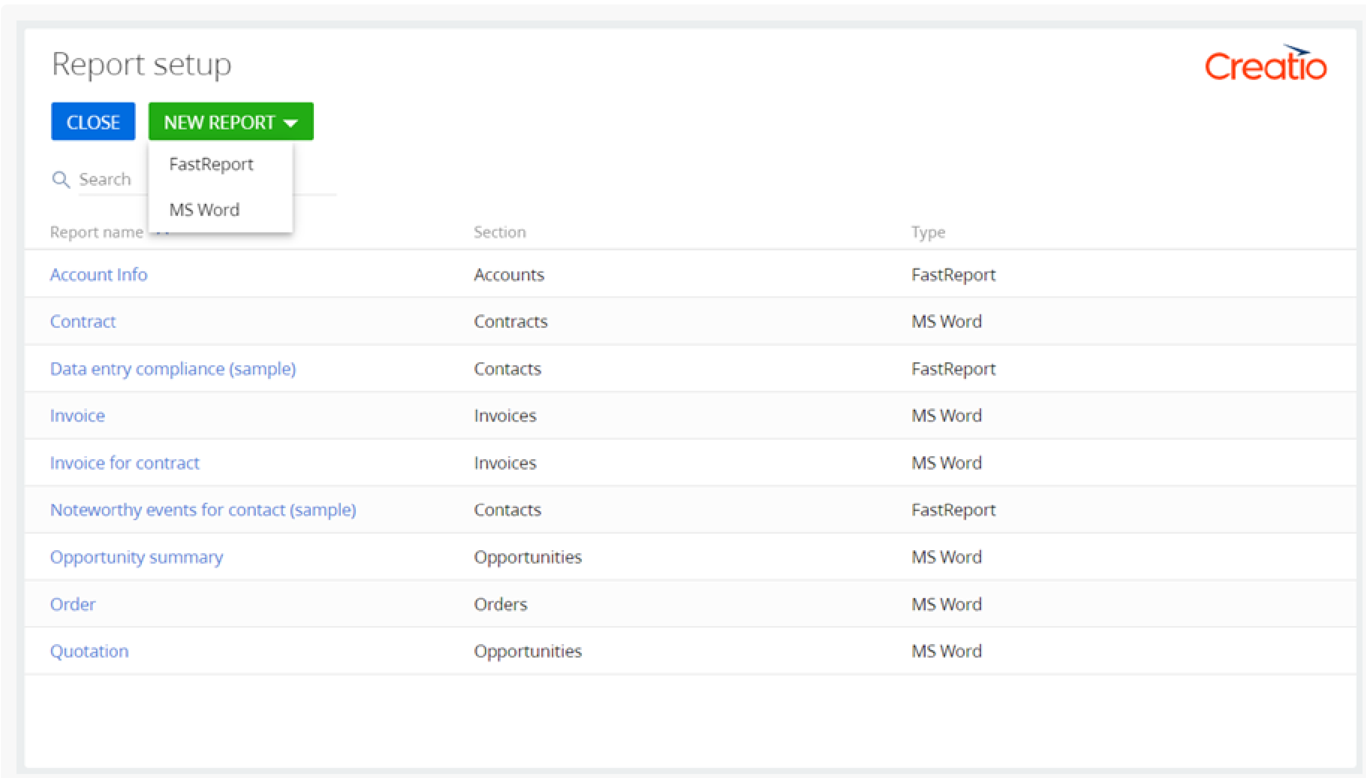
- [*Type*].
- [*Primary contact*].

Case implementation algorithm

1. Create a new report

To do this:

1. Open the System Designer by clicking . In the [*System setup*] block, click the [*Report setup*] link.
2. Click [*New report*] –> [*MS Word*].

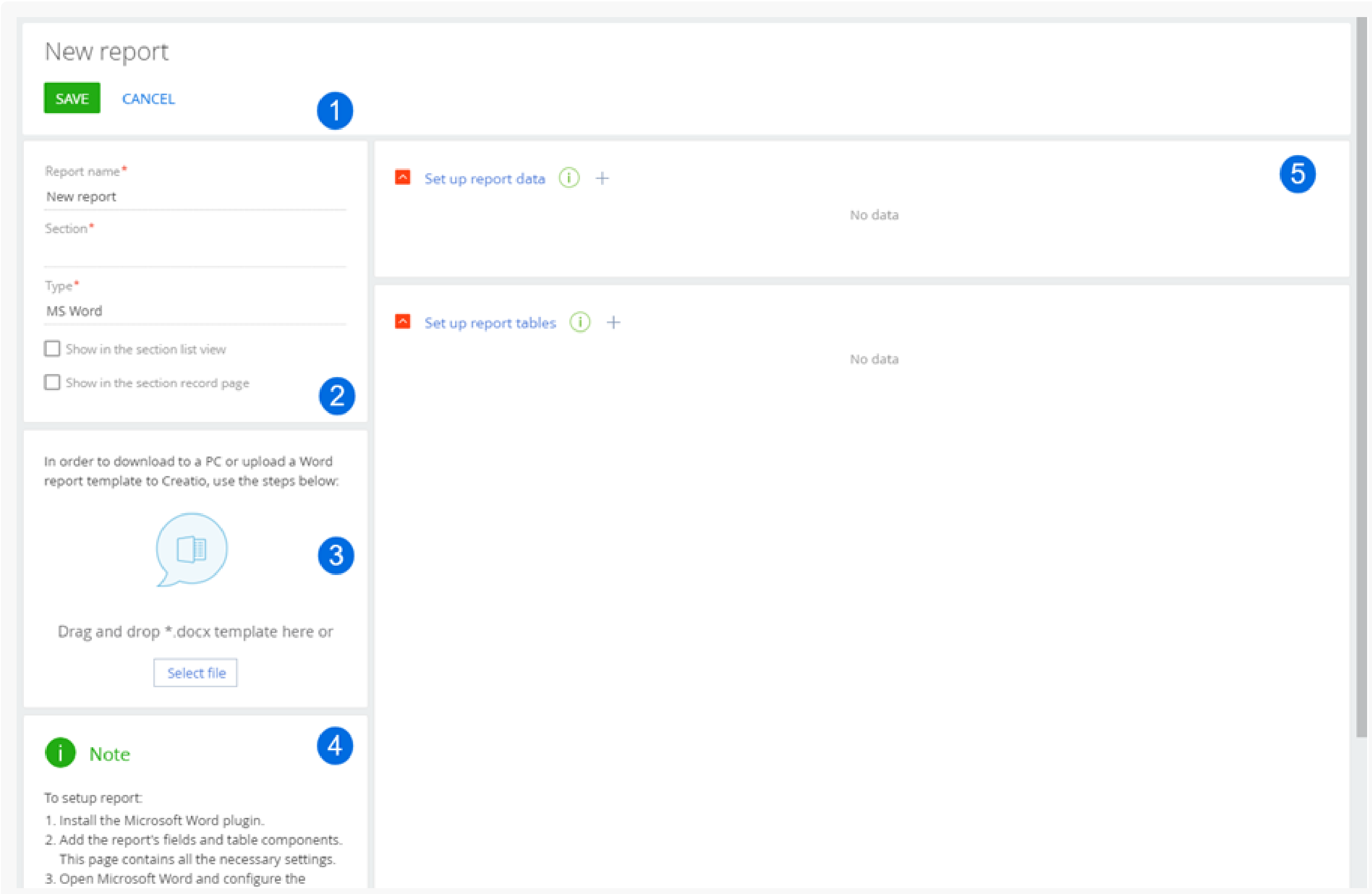


2. Set up the report display parameters

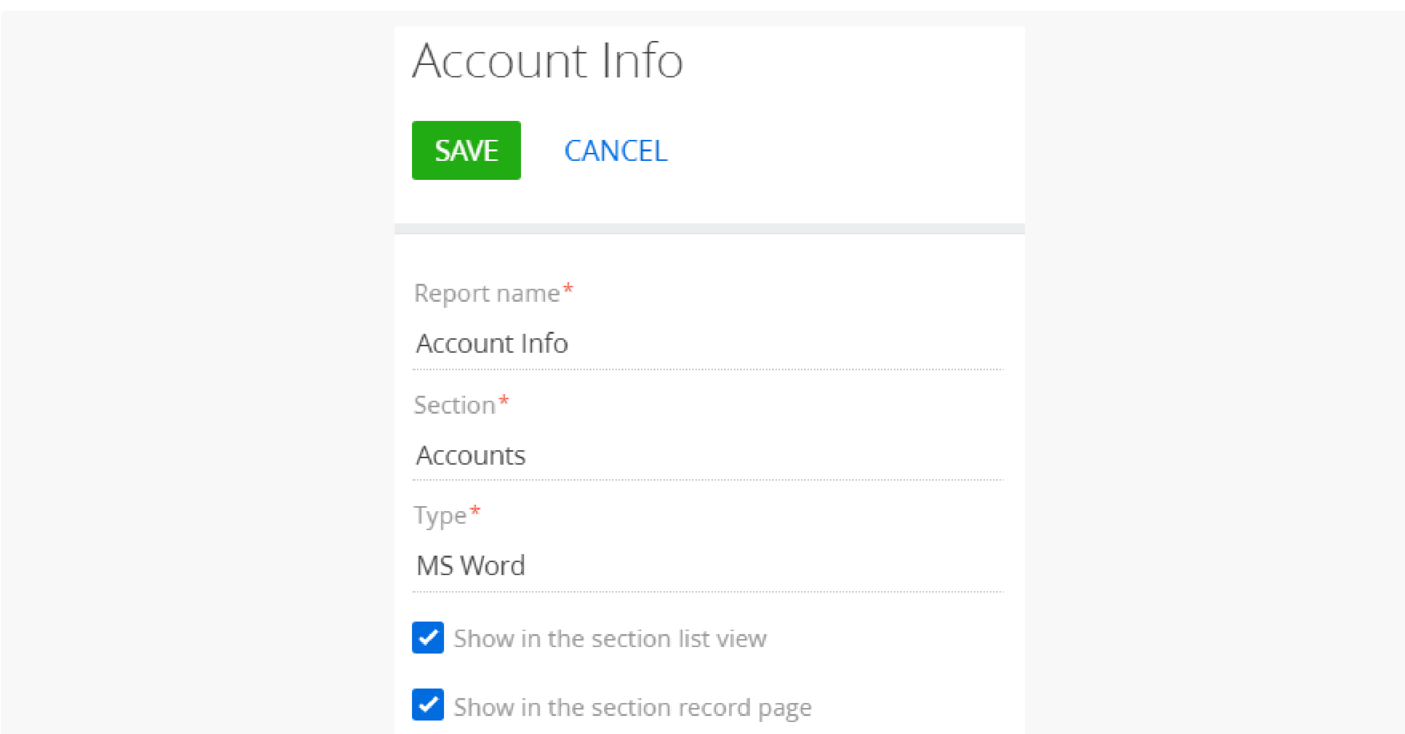
Set the following values in the parameter setup area (2):

- [*Report title*] – "Account Info".
- [*Section*] – "Accounts".
- [*Show in the section list view*].
- [*Show in the section record page*].

MS Word report setup page

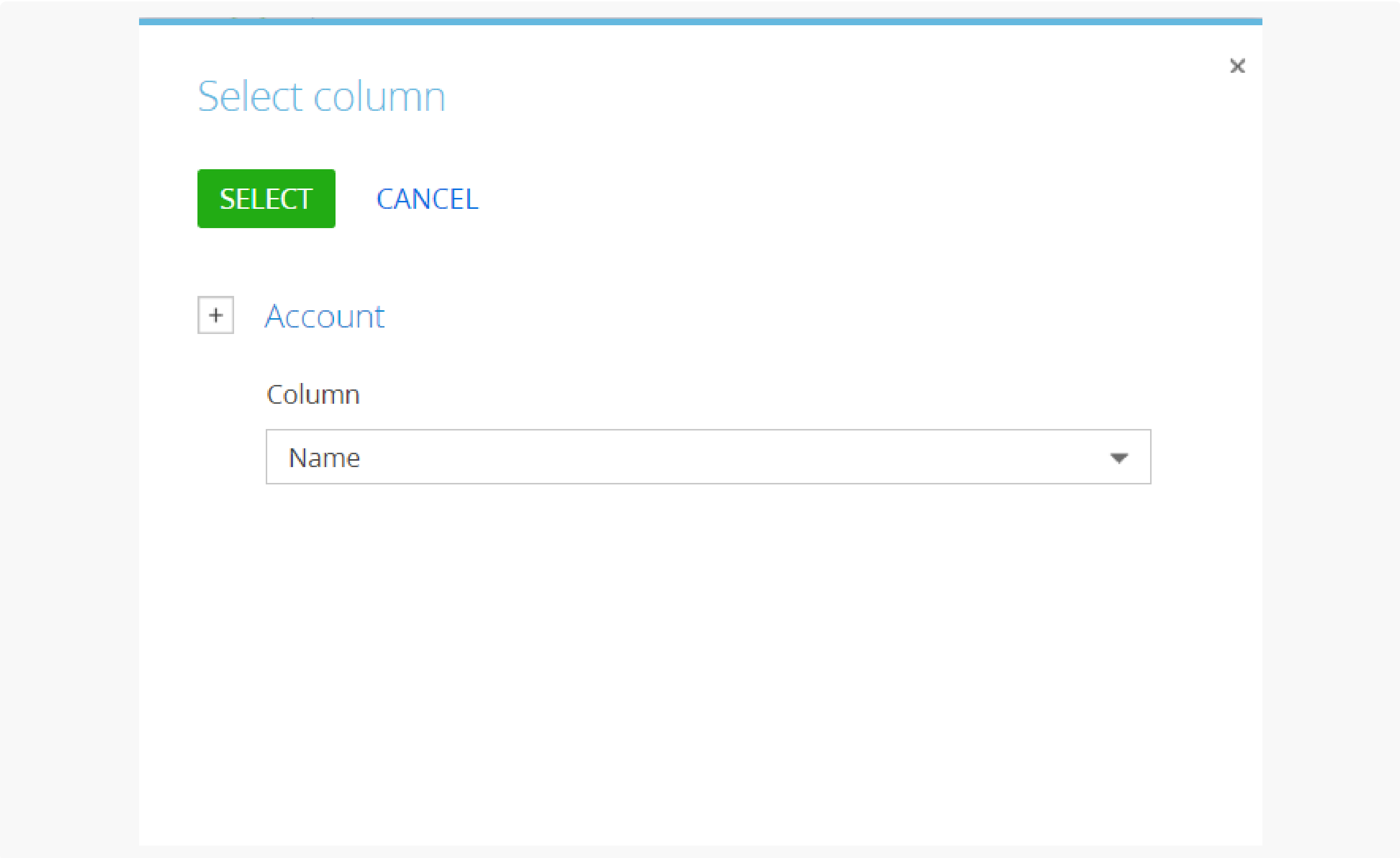


Setting up the report display parameters



3. Set up the report fields

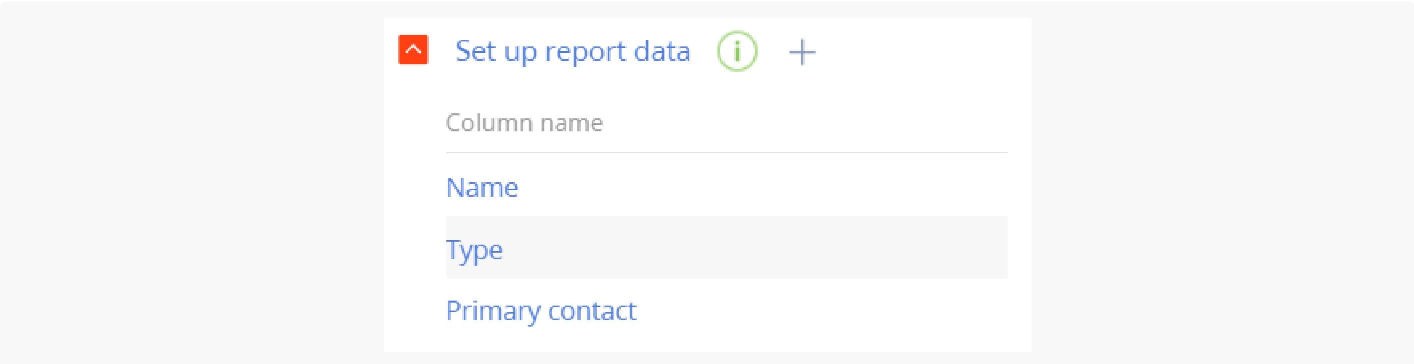
In the [*Set up report data*] block of the section working area (5), set up the fields to display in the report. To do this, click + and select the [*Name*] column in the drop-down [*Column*] list. A macro will be added to the column later.



Click [*Select*].

Using the same method, add the [*Type*] and [*Primary contact*] columns to the template.

The list of columns after this step is presented below.



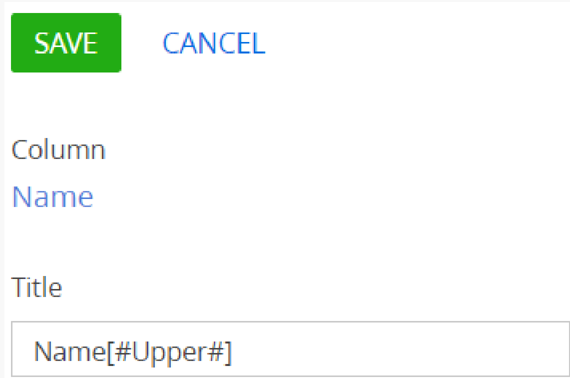
4. Add the macro tag to the column name

Change the [*Name*] column property. To do this, take the following steps;

1. In the [*Set up report data*] block of the section working area (5), double-click the title of the [*Name*] column

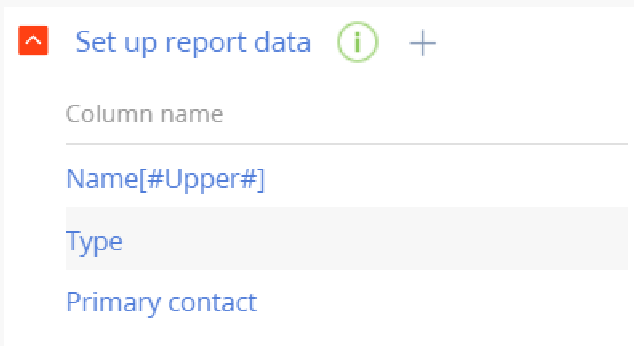
or click in the column title bar.

2. Change the `[Name]` value of the [*Title*] field to `[Name[#Upper#]]`.



Click [*Save*].

The list of columns after adding macro tags is presented below .



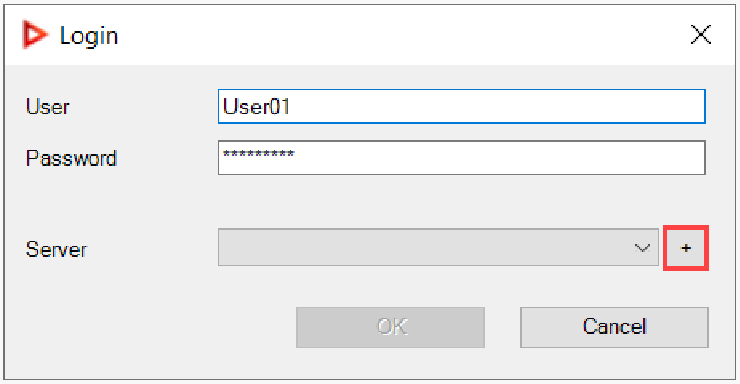
Click [*Save*].

5. Set up the report template layout and upload the template to Creatio

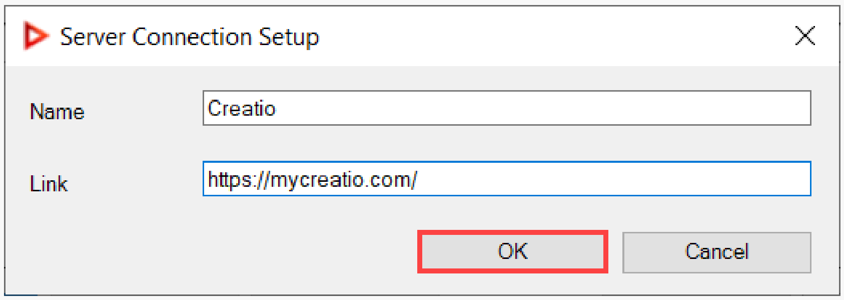
To set up the template:

1. Open any MS Word file.
2. Click [*Connect*] on the Creatio plug-in toolbar.

3. Enter the username and password of the Creatio user. Click + next to the [*Server*] field.

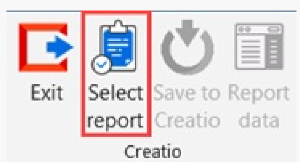


4. Click [*New*]. Enter the server parameters.

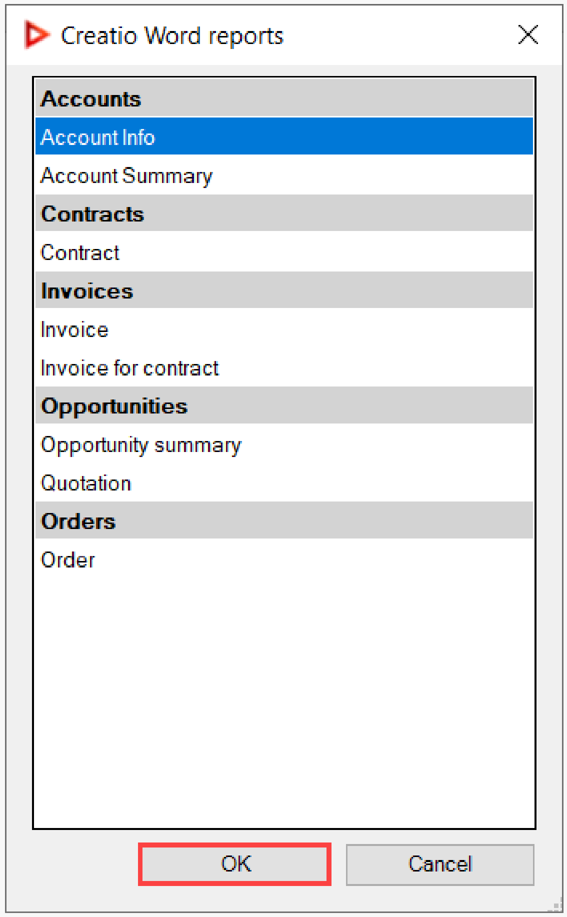


Click [*OK*].

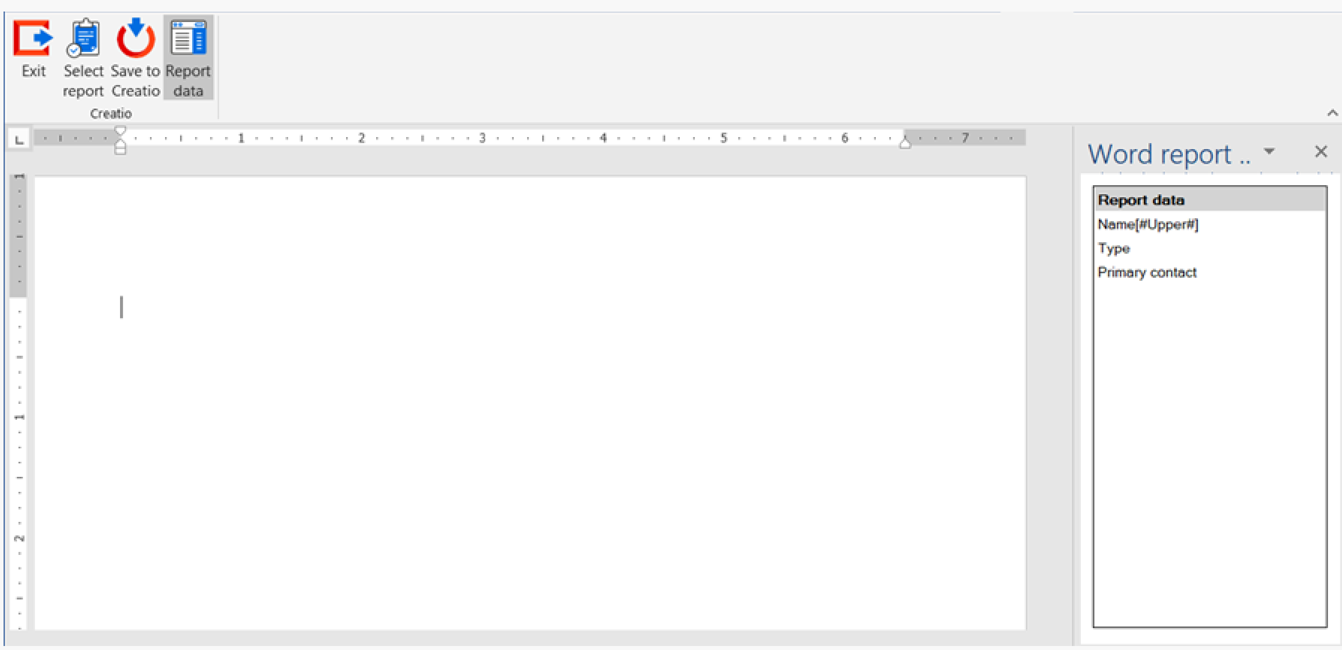
5. Click [*Select report*] on the Creatio plug-in toolbar.



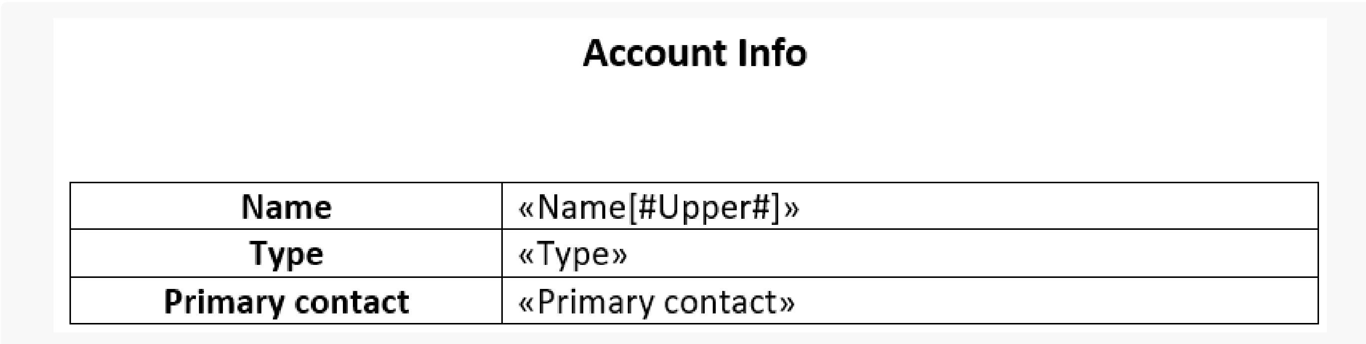
6. Select the "Account info" report and click [*OK*].



The report setup window looks as follows:



7. Set up the template layout. Learn more about setting up a report template in the "[Design report layout via the Creatio MS Word plug-in]()" article.
After the setup, the report looks as follows:

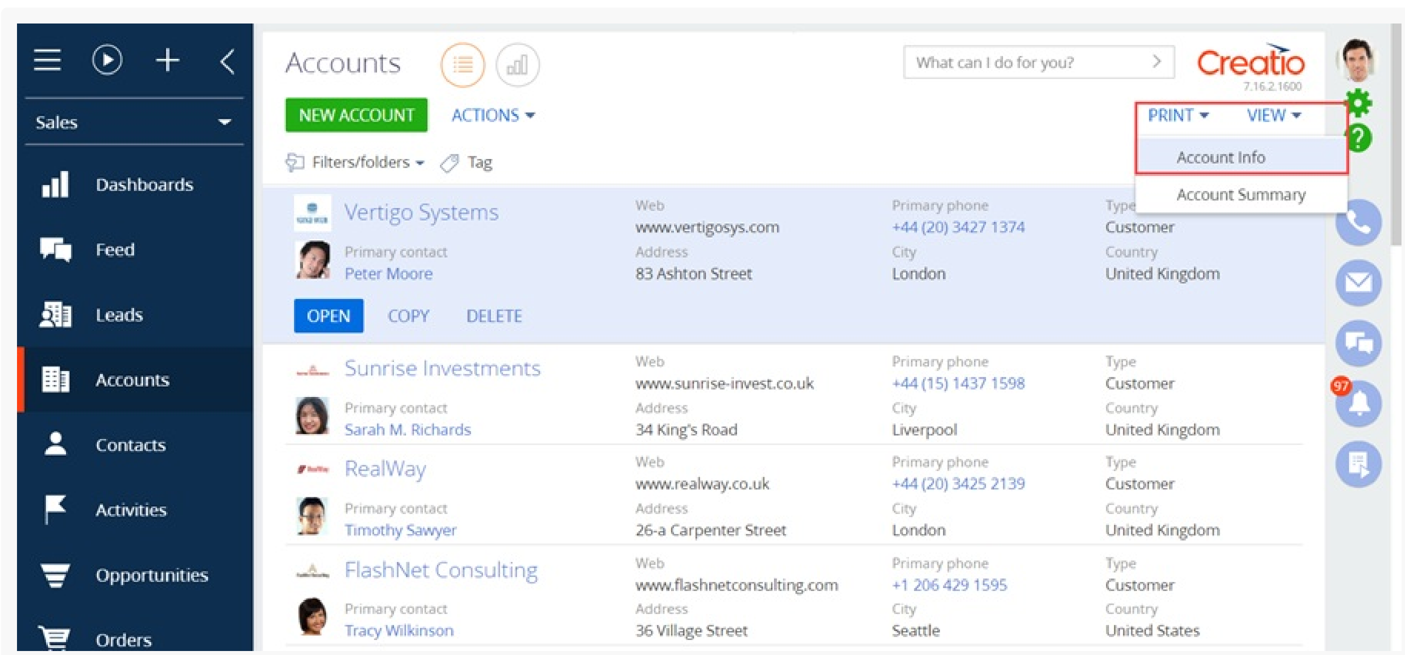


Account Info

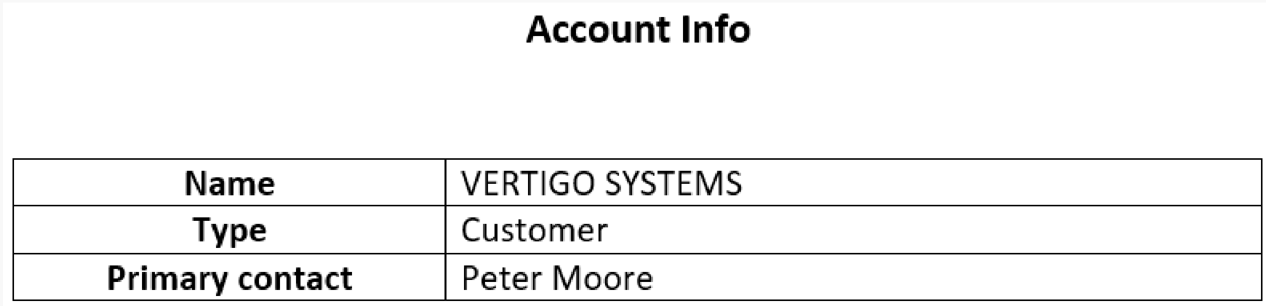
Name	«Name[#Upper#]»
Type	«Type»
Primary contact	«Primary contact»

8. Click [*Save to Creatio*] to load the report template in Creatio.

As a result, the "Account Info" report will be available in the [*Accounts*] section list under [*Print*].



The report looks as follows.



Account Info

Name	VERTIGO SYSTEMS
Type	Customer
Primary contact	Peter Moore

Create MS Word report using custom macros

Advanced

Case. Create an "Account Summary" report for the [*Accounts*] section edit page to display the following information about the account:

- [*Name*].
- [*Type*].
- [*Primary contact*].
- [*Additional info*]. The annual revenue should be displayed for [*Customer*] accounts and the number of employees for [*Partner*] accounts.

The report must contain information about the date of creation and the name of the employee who created it.

Source code

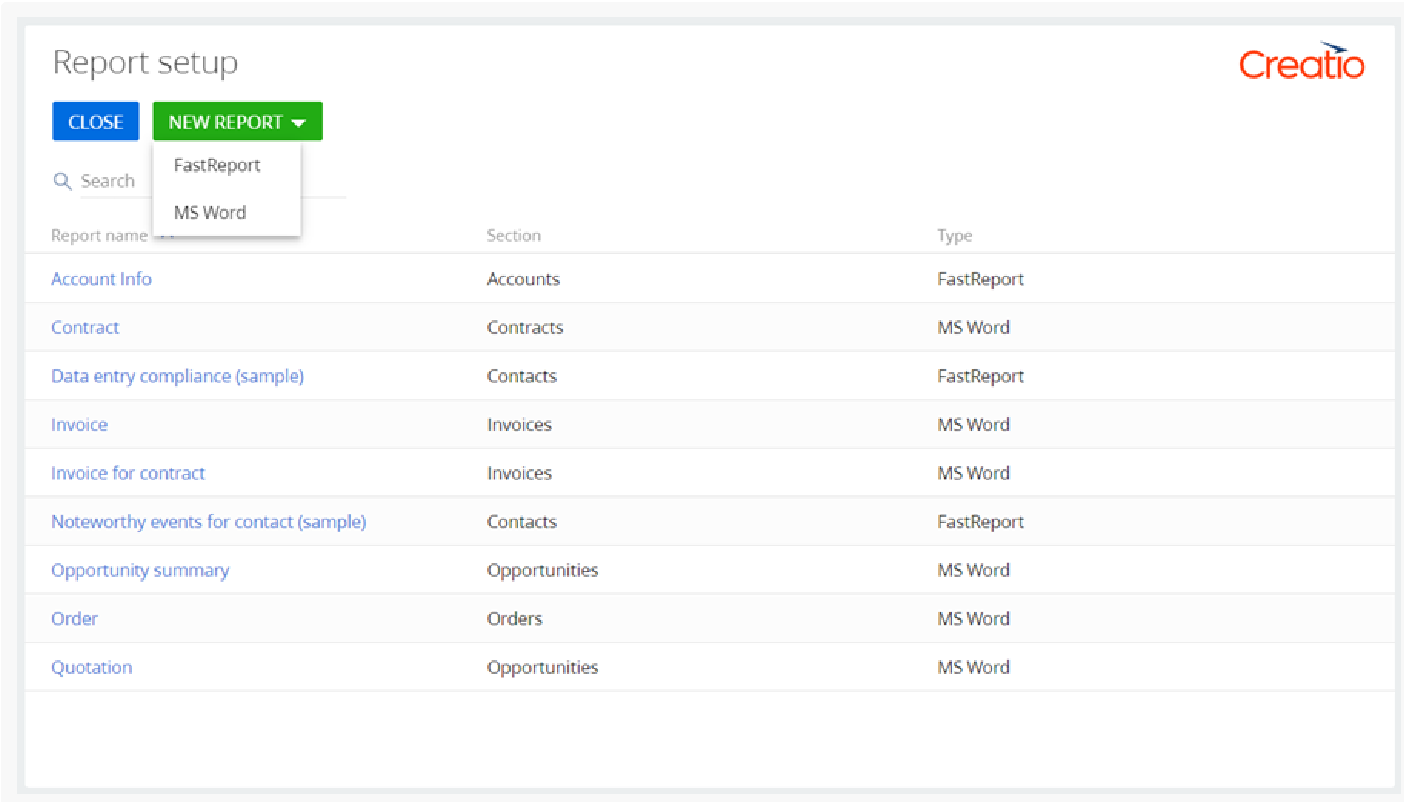
You can download the package with an implementation of the case using the following link.

Case implementation algorithm

1. Create a new report

To do this:

1. Open the System Designer by clicking . In the [*System setup*] block, click the [*Report setup*] link.
2. Click [*New report*] —>[*MS Word*].

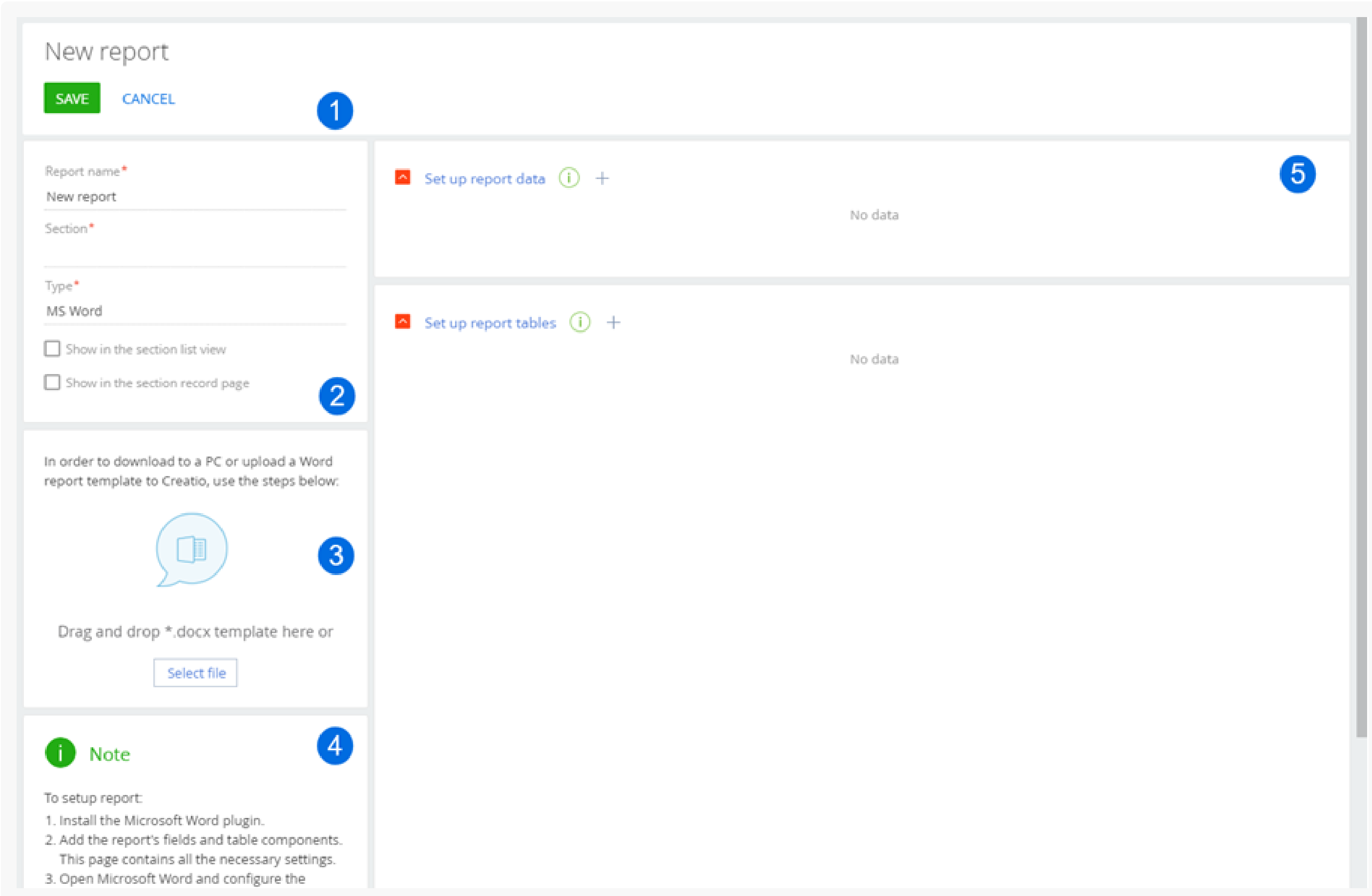


2. Set up the report display parameters

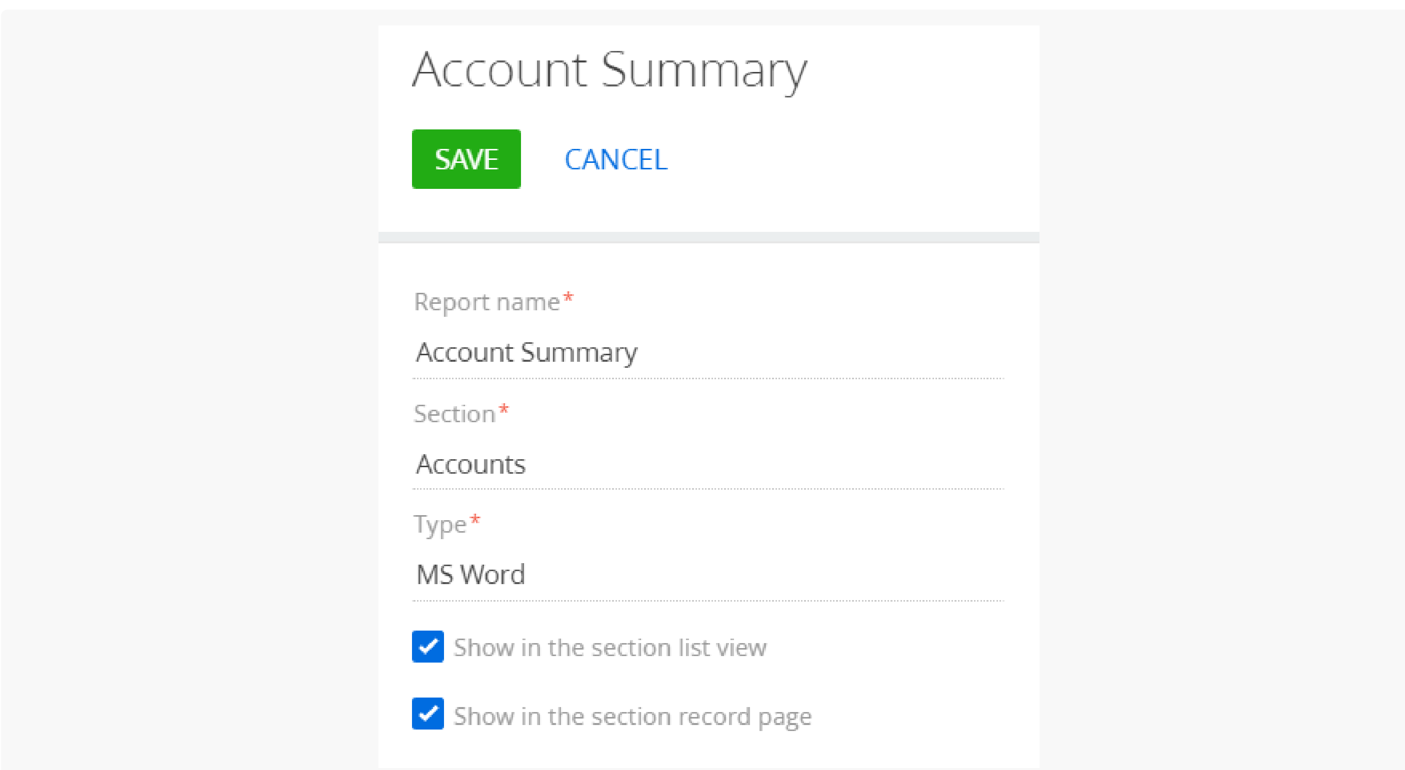
Set the following values in the parameter setup area (2):

- [*Report title*] – "Account Summary".
- [*Section*] – "Accounts".
- [*Show in the section list view*].
- [*Show in the section record page*].

MS Word report setup page

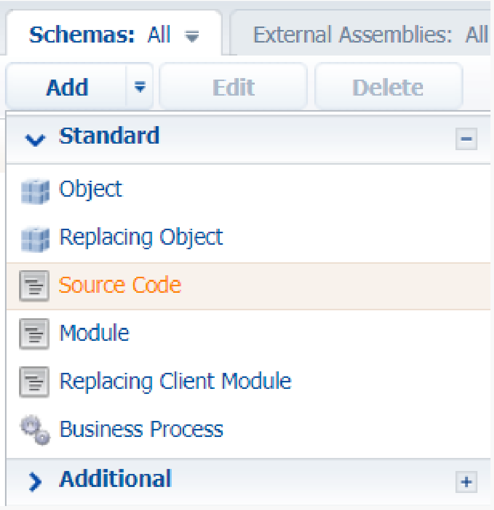


Setting up the report display parameters



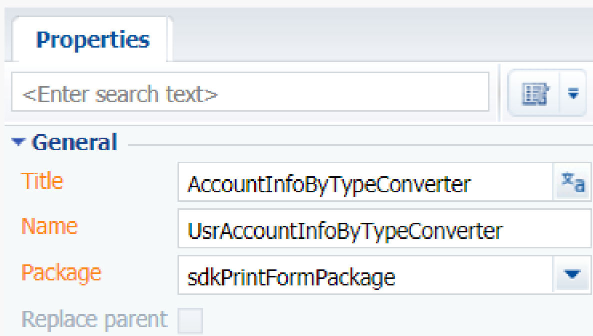
3. Implement custom macros

Go to the [Advanced settings] section –> [*Configuration*] –> Custom package –> the [*Schemas*] tab. Click [*Add*] —> [*Source Code*]. Learn more about creating a schema of the [*Source Code*] type in the [Create the [*Source code*] schema]" article.



Specify the following parameters for the created object schema:

- [*Title*] – "AccountInfoByTypeConverter".
- [*Name*] – "UsrAccountInfoByTypeConverter".



Implement a macro class for receiving additional information depending on the account type. The complete source code of the module is available below:

```
namespace Terrasoft.Configuration
{
    using System;
    using System.CodeDom.Compiler;
    using System.Collections.Generic;
    using System.Data;
    using System.Linq;
    using System.Runtime.Serialization;
    using System.ServiceModel;
    using System.ServiceModel.Web;
    using System.ServiceModel.Activation;
    using System.Text;
```

```
using System.Text.RegularExpressions;
using System.Web;
using Terrasoft.Common;
using Terrasoft.Core;
using Terrasoft.Core.DB;
using Terrasoft.Core.Entities;
using Terrasoft.Core.Packages;
using Terrasoft.Core.Factories;

// An attribute with the [AccountInfoByType] macro name.
[ExpressionConverterAttribute("AccountInfoByType")]
// The class should implement the IExpressionConverter interface.
class AccountInfoByTypeConverter : IExpressionConverter
{
    private UserConnection _userConnection;
    private string _customerAdditional;
    private string _partnerAdditional;
    // Calling localizable string values
    private void SetResources() {
        string sourceCodeName = "UsrAccountInfoByTypeConverter";
        _customerAdditional = new LocalizableString(_userConnection.ResourceStorage, sourceC
            "LocalizableStrings.CustomerAdditional.Value");
        _partnerAdditional =  new LocalizableString(_userConnection.ResourceStorage, sourceC
            "LocalizableStrings.PartnerAdditional.Value");
    }
    // Implementing the Evaluate method of the IExpressionConverter interface.
    public string Evaluate(object value, string arguments = "")
    {
        try
        {
            _userConnection = (UserConnection)HttpContext.Current.Session["UserConnection"];
            Guid accountId = new Guid(value.ToString());
            return getAccountInfo(accountId);
        }
        catch (Exception err)
        {
            return err.Message;
        }
    }
    // The method for receiving additional information depending on the account type.
    // As the Id input parameter of the account.
    private string getAccountInfo(Guid accountId)
    {
     SetResources();
        try
        {
            // Creating an EntitySchemaQuery class instance with the [Account] root schema.
            EntitySchemaQuery esq = new EntitySchemaQuery(_userConnection.EntitySchemaManage
            // Adding the [Name] column from the [Type] lookup field.
```

```
                var columnType = esq.AddColumn("Type.Name").Name;
                // Adding the [Name] column from the [EmployeesNumber] lookup field.
                var columnNumber = esq.AddColumn("EmployeesNumber.Name").Name;
                // Adding the [Name] column from the [AnnualRevenue] lookup field.
                var columnRevenue = esq.AddColumn("AnnualRevenue.Name").Name;
                // The records are filtered by the account Id.
                var accountFilter = esq.CreateFilterWithParameters(
                    FilterComparisonType.Equal,
                    "Id",
                    accountId
                );
                esq.Filters.Add(accountFilter);
                // Retrieving an entity collection.
                EntityCollection entities = esq.GetEntityCollection(_userConnection);
                // If the collection is not empty, the method will return the corresponding
                // data depending on the account
                if (entities.Count > 0)
                {
                    Entity entity = entities[0];
                    var type = entity.GetTypedColumnValue(columnType);
                    switch (type)
                    {
                        case "Customer":
                            return String.Format(_customerAdditional, entity.GetTypedColumnValue
                        case "Partner":
                            return String.Format(_partnerAdditional, entity.GetTypedColumnValue(
                        default:
                            return String.Empty;
                    }
                }
                return String.Empty;
            }
            catch (Exception err)
            {
                throw err;
            }
        }
    }
}
```

Populate the localizable strings of the report with the following values:

Setting up the localizable strings

Name	English (United States)	Russian (Russia)
PartnerAdditional	Number of employees {0} persons	Number of employees {0} people

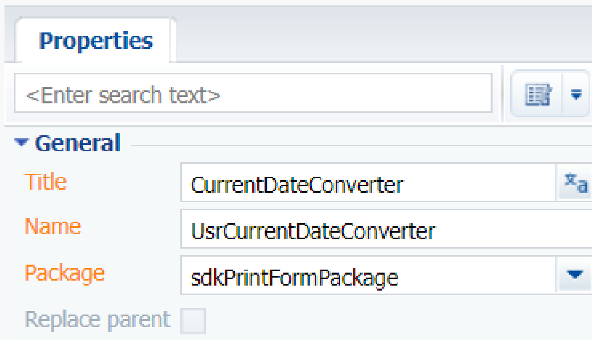
CustomerAdditional	Annual turnover {0}	Annual revenue {0}

After making changes, save and publish the schema.

Go to the [*Advanced settings*] section -> [*Configuration*] -> Custom package -> the [*Schemas*] tab. Click [*Add*] —> [*Source Code*].

Specify the following parameters for the created object schema:

- [*Title*] – "CurrentDateConverter".
- [*Name*] – "UsrCurrentDateConverter".



Implement a macro class for retrieving the current date. The complete source code of the module is available below:

```
namespace Terrasoft.Configuration
{
    using System;
    using System.CodeDom.Compiler;
    using System.Collections.Generic;
    using System.Data;
    using System.Linq;
    using System.Runtime.Serialization;
    using System.ServiceModel;
    using System.ServiceModel.Web;
    using System.ServiceModel.Activation;
    using System.Text;
    using System.Text.RegularExpressions;
    using System.Web;
    using Terrasoft.Common;
    using Terrasoft.Core;
    using Terrasoft.Core.DB;
    using Terrasoft.Core.Entities;
    using Terrasoft.Core.Packages;
    using Terrasoft.Core.Factories;

    // An attribute with the [CurrentDate] macro name.
    [ExpressionConverterAttribute("CurrentDate")]
    // The class should implement the IExpressionConverter interface.
    class CurrentDateConverter : IExpressionConverter
```

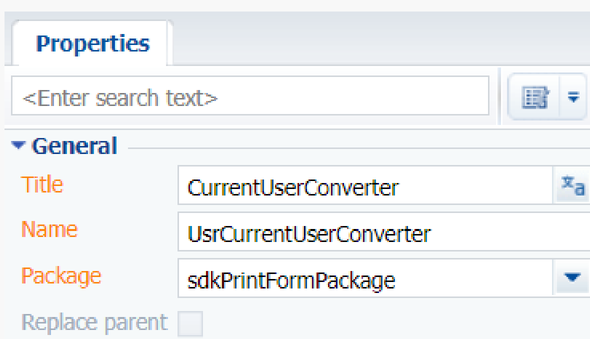
```
    {
        private UserConnection _userConnection;

        // Implementing the Evaluate method of the IExpressionConverter interface.
        public string Evaluate(object value, string arguments = "")
        {
            try
            {
                _userConnection = (UserConnection)HttpContext.Current.Session["UserConnection"];
                // The method returns the current date.
                return _userConnection.CurrentUser.GetCurrentDateTime().Date.ToString("dd MMM yy
            }
            catch (Exception err)
            {
                return err.Message;
            }
        }
    }
}
```

After making changes, save and publish the schema.

Go to the [*Advanced settings*] section -> [*Configuration*] -> Custom package -> the [*Schemas*] tab. Click [*Add*] —> [*Source Code*].

Specify the following parameters for the created object schema:

- [*Title*] – "CurrentUserConverter"
- [*Name*] – "UsrCurrentUserConverter".



Implement a macro class for retrieving the current user. The complete source code of the module is available below:

```
namespace Terrasoft.Configuration
{
    using System;
    using System.CodeDom.Compiler;
    using System.Collections.Generic;
    using System.Data;
```

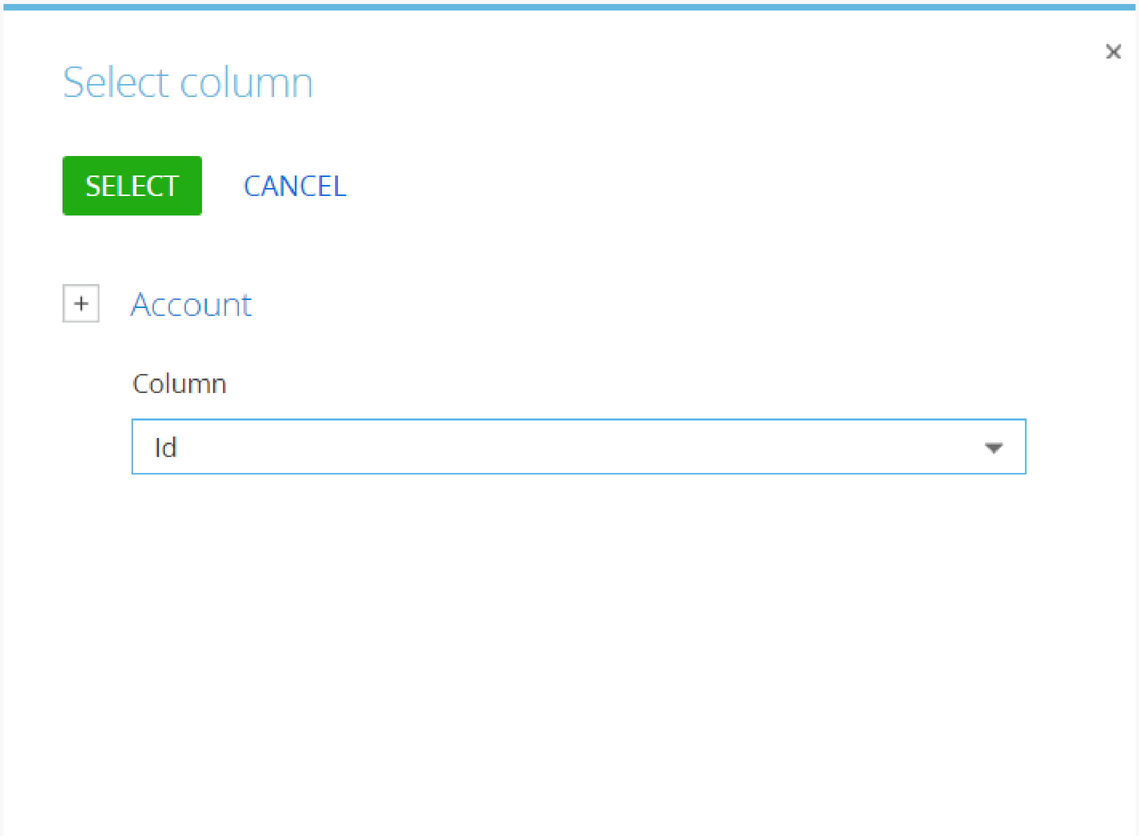
```
    using System.Linq;
    using System.Runtime.Serialization;
    using System.ServiceModel;
    using System.ServiceModel.Web;
    using System.ServiceModel.Activation;
    using System.Text;
    using System.Text.RegularExpressions;
    using System.Web;
    using Terrasoft.Common;
    using Terrasoft.Core;
    using Terrasoft.Core.DB;
    using Terrasoft.Core.Entities;
    using Terrasoft.Core.Packages;
    using Terrasoft.Core.Factories;

    // An attribute with the [CurrentUser] macro name.
    [ExpressionConverterAttribute("CurrentUser")]
    // The class should implement the IExpressionConverter interface.
    class CurrentUserConverter : IExpressionConverter
    {
        private UserConnection _userConnection;
        // Implementing the Evaluate method of the IExpressionConverter interface.
        public string Evaluate(object value, string arguments = "")
        {
            try
            {
                _userConnection = (UserConnection)HttpContext.Current.Session["UserConnection"];
                // The method returns the contact of the current user.
                return _userConnection.CurrentUser.ContactName;
            }
            catch (Exception err)
            {
                return err.Message;
            }
        }
    }
}
```

After making changes, save and publish the schema.

4. Set up the report fields

In the [*Set up report data*] block of the section working area (5), set up the fields to display in the report. To do this, click + and select the [*Id*] column in the drop-down [*Column*] list. The current [*Id*] column will later be used in the custom macro to retrieve the current date.

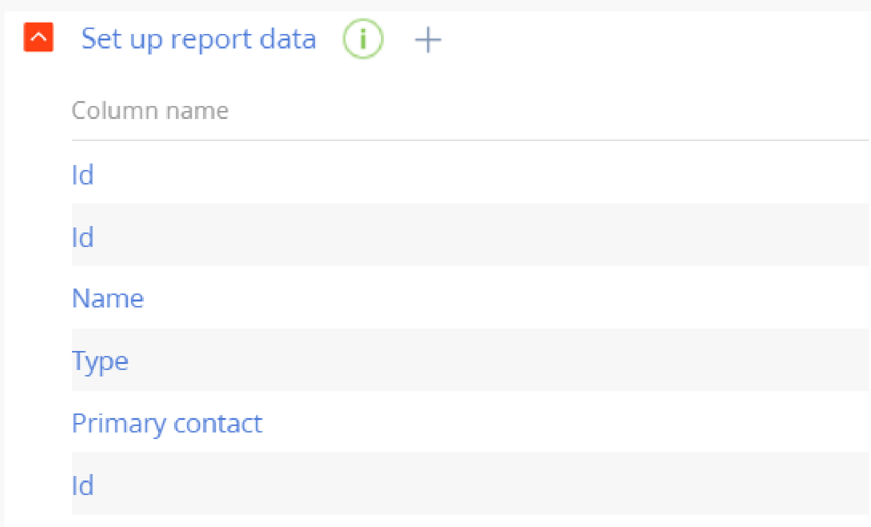
Attention. Use the [*Id*] column as an input parameter for a custom macro.



Click [*Select*].

Use the same procedure to add [*Id*] (the column will later be used in the custom macro for retrieving the current user), [*Name*], [*Type*], [*Primary contact*], and [*Id*] (the column will later be used in the custom macro for receiving additional information depending on the account type) to the column template.

The list of columns after this step is presented below.

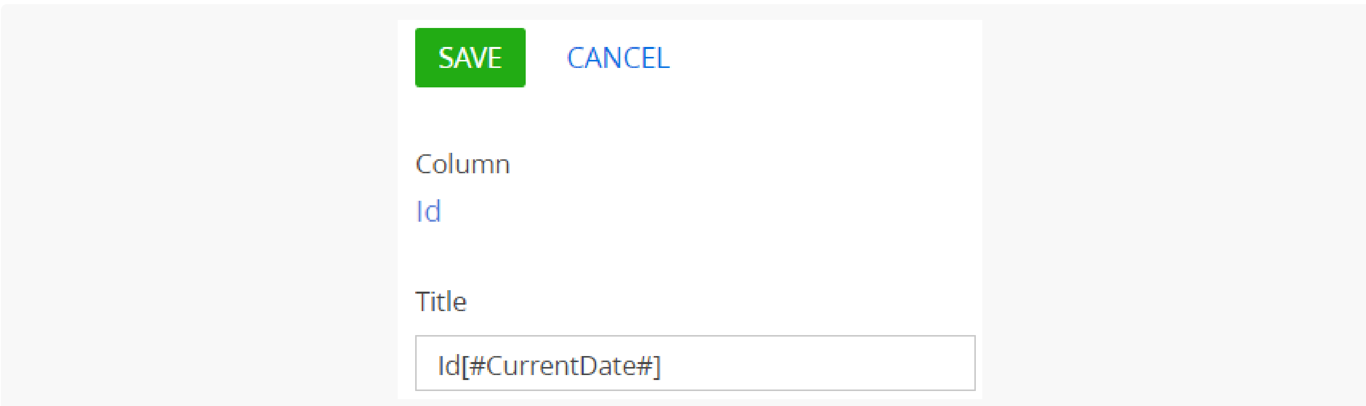


5. Attach custom macro tags to the column names

Attention. First, publish the [*Source Code*] type schema that implements a custom macro must. Then, add the name of the macro to the template layout. If you refresh the page in Creatio, the macro will not be printed.

Change the property of the [*Id*] column of an [*Account*] object. To do this, take the following steps:

1. In the [*Set up report data*] block of the section working area (5), double-click the title of the [*Id*] column or click in the column title bar.
2. Change the `[Id]` value of the [*Title*] field to `[Id[#CurrentDate#]]`. `[#CurrentDate#]` is a the custom macro tag for retrieving the current date.

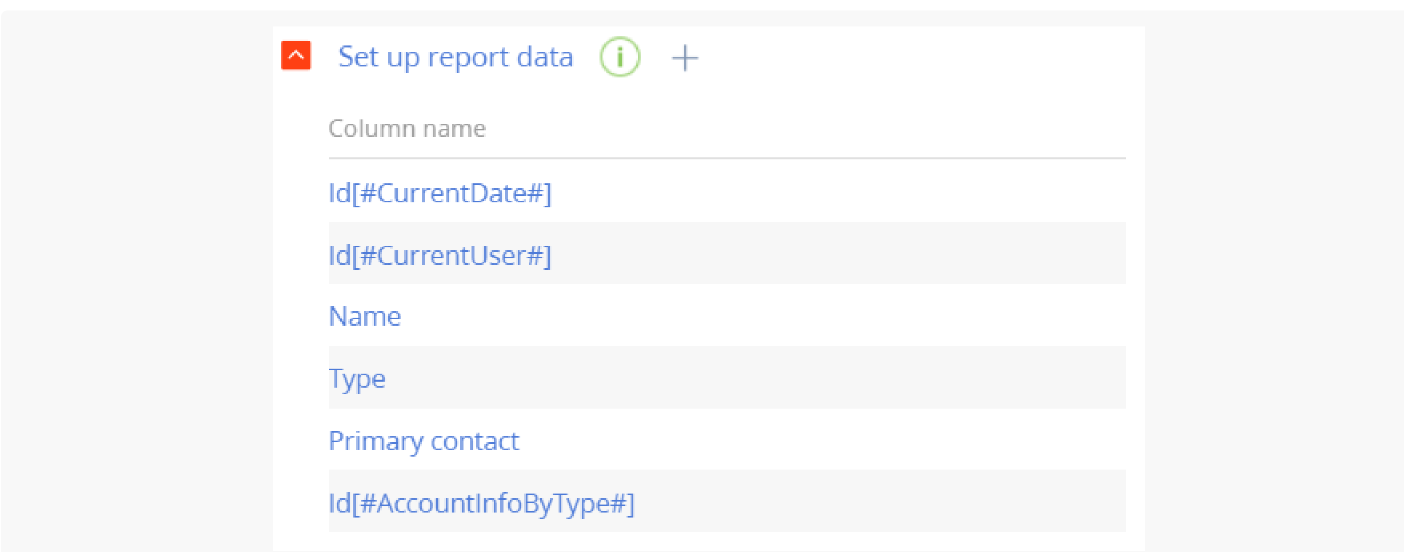


Click [*Save*].

Use the same procedure to add more custom macro tags to the names of other [*Id*] columns.

- `[#CurrentUser]#` – for receiving the current user.
- `[#AccountInfoByType#]` – for receiving additional information depending on the account type.

The list of columns after adding custom macro tags is presented below.

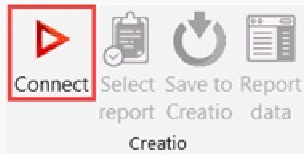


Click [*Save*].

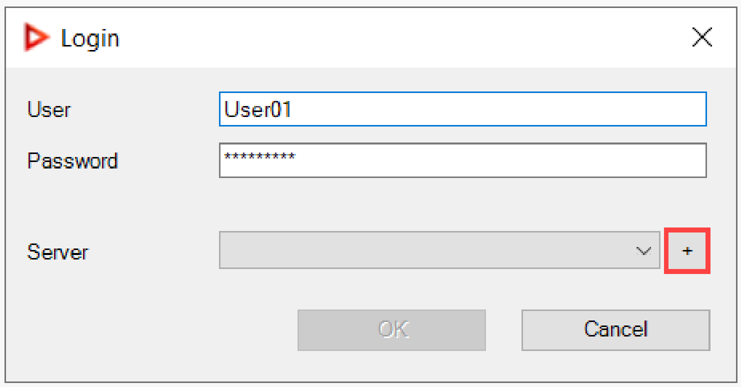
6. Set up the report template layout and upload the template to Creatio

To set up the template:

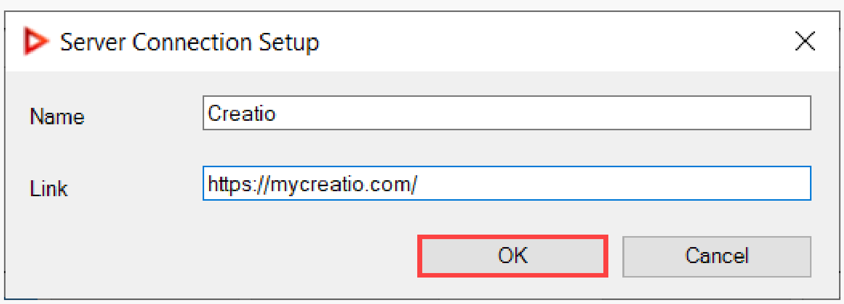
1. Open any MS Word file.
2. Click [*Connect*] on the Creatio plug-in toolbar.



3. Enter the username and password of the Creatio user. Click + next to the [*Server*] field.

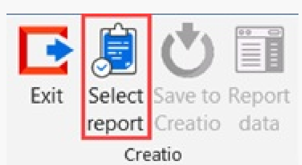


4. Click [*New*]. Enter the server parameters.

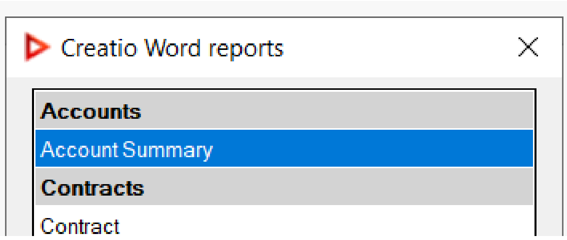


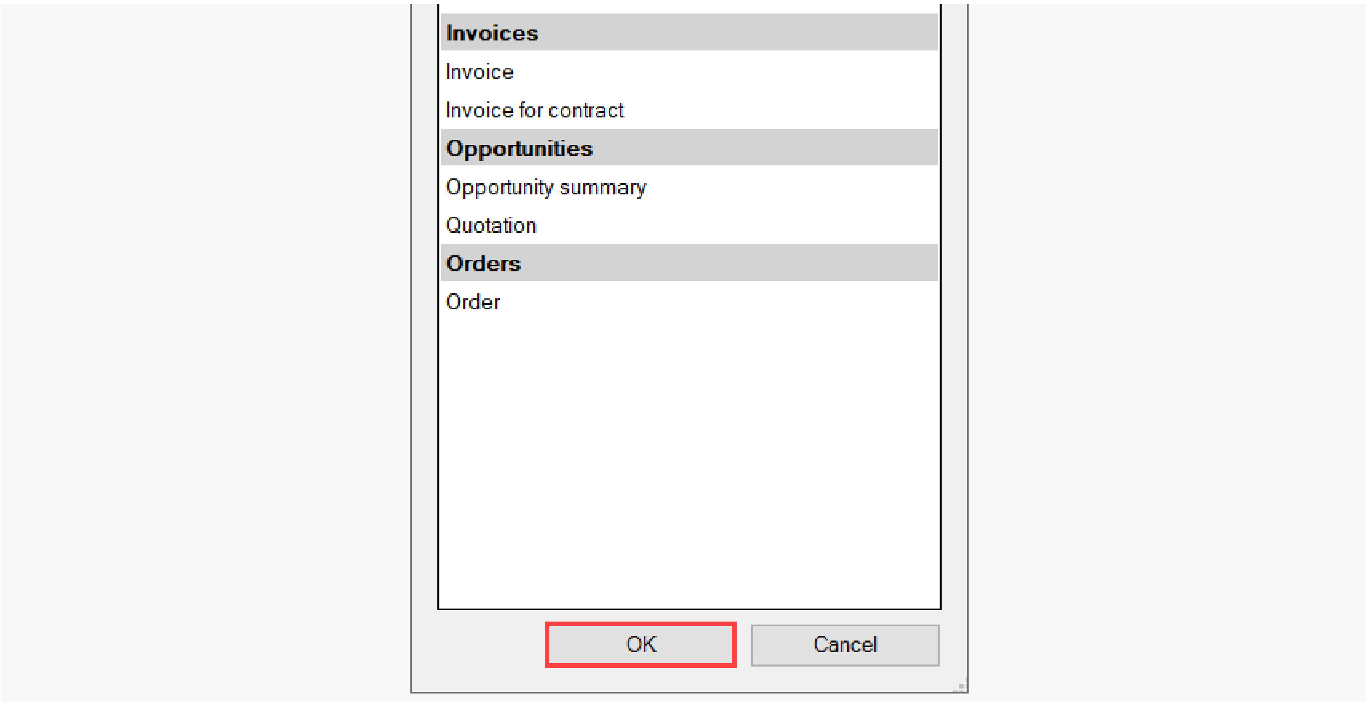
Click [*OK*].

5. Click [*Select report*] on the Creatio plug-in toolbar.

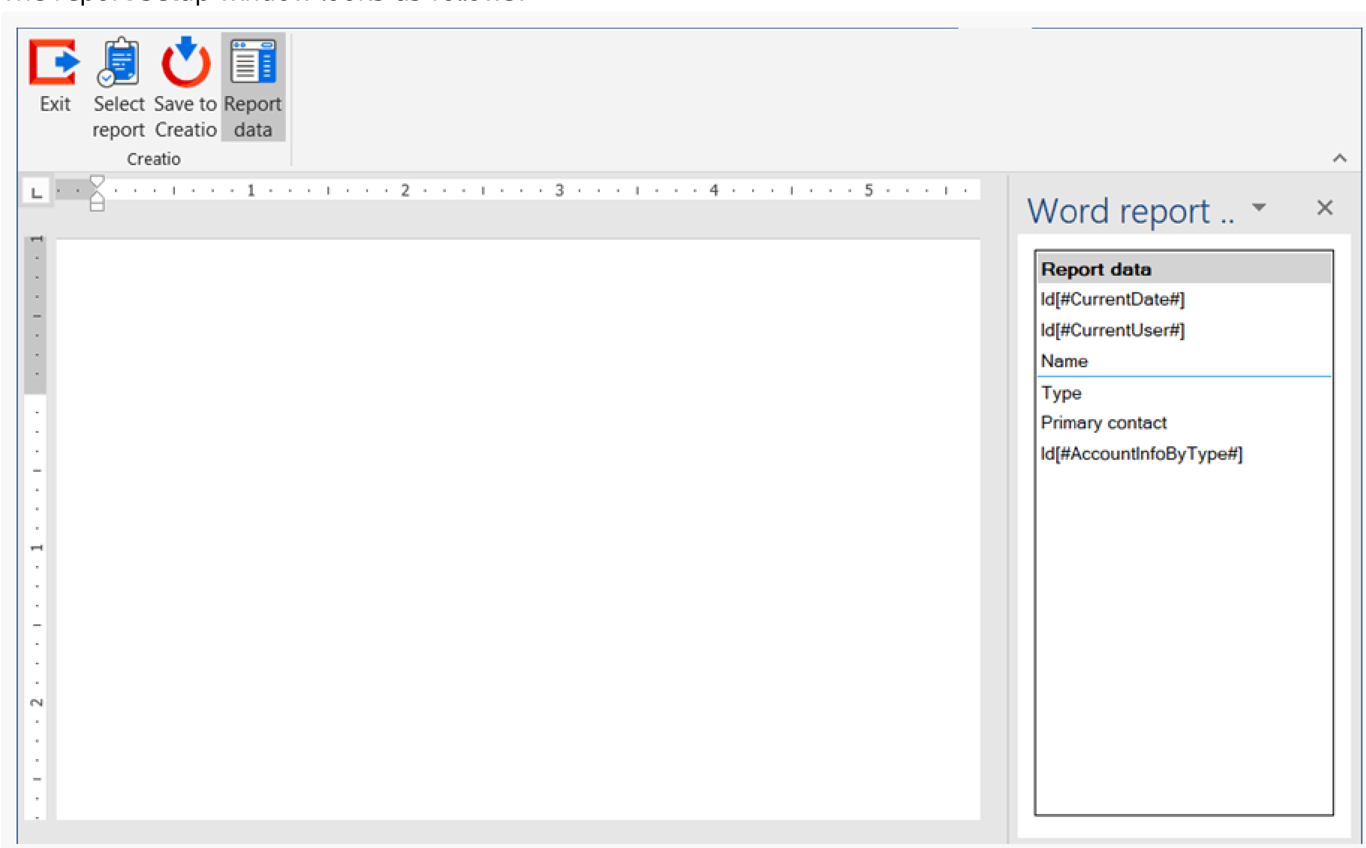


6. Select the "Account summary" and click [*OK*].





The report setup window looks as follows:



7. Set up the template layout. Learn more about setting up a report template in the "[Design report layout via the Creatio MS Word plug-in]()" article.

   After the setup, the report looks as follows:

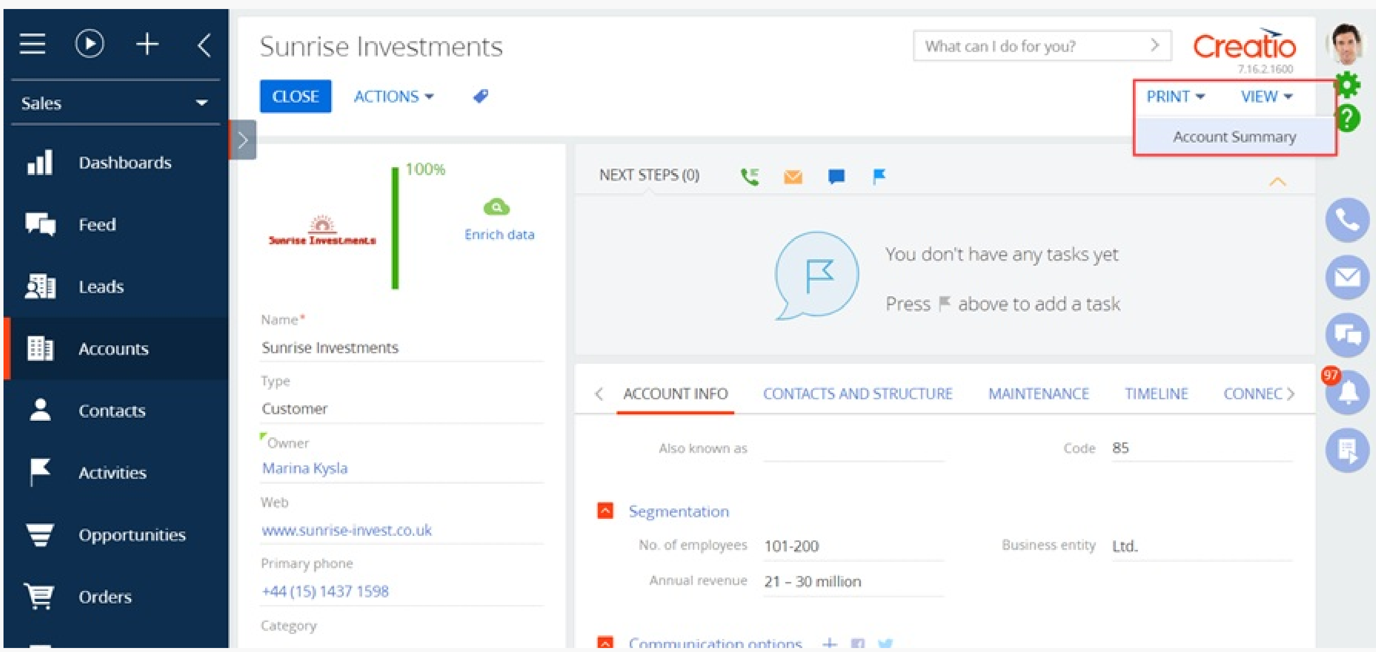
Name	«Name»
Type	«Type»
Primary contact	«Primary contact»
Additional info	«Id[#AccountInfoByType#]»

Date «Id[#CurrentDate#]»

Created by «Id[#CurrentUser#]»

8. Click [*Save to Creatio*] to load the configured report template in Creatio.

As a result, the "Account Summary" report will be available on the contact page under [*Print*].



A report for accounts of the [*Customer*] type looks as follows.

Account Summary

Name	Sunrise Investments
Type	Customer
Primary contact	Sarah M. Richards
Additional info	Annual turnover 21 – 30 million

Additional info	Annual turnover 21 – 30 million

Date 14 Jul 2020

Created by User01

A report for accounts of the [*Partner*] type looks as follows.

Account Summary

Name	ClearSoft Systems
Type	Partner
Primary contact	William Clarks
Additional info	Number of employees 51-100 persons

Date 14 Jul 2020

Created by User01