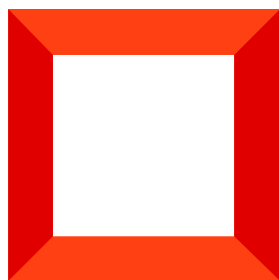
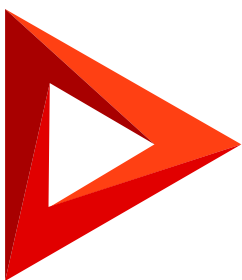


Emails

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Table of Contents

Work with emails basics	4
Junk case custom filtering	4
Working with email threads	4
Macro handler in the email template	8
Add a email filtering property	8
Case description	8
Case implementation algorithm	8
Add the macro handler to the email template	9
Case description	9
Source code	9
Case implementation algorithm	9
Sending emails	12
Sending an email from an existing account algorithm	12
Sending an email using the explicit account credentials algorithm	14
Send email from an existing account	16
Example implementation algorithm	16
Send email using the explicit account credentials	26
Example implementation algorithm	26

Work with emails basics



Junk case custom filtering

Creatio users can filter unwanted cases by creating a list of email addresses and domains for automatic spam detection. Incoming cases from these domains or addresses are either not registered at all or have the “Canceled” status (the default status value is set in the [*Junk Case Default Status*] system setting).

Junk filter enables email header analysis based on certain flags, e.g. `Auto-Submitted`. Emails with those flags are either not registered or register with the pre-defined initial status, set in system settings.

To analyze email addresses and domains, simply add the required value to the [*Blacklist of email addresses and domains for case registration*] lookup. The emails will be marked as blacklisted during the analysis when you populate the lookup with values (their type is set automatically).

Email analysis is done through the [*Email header properties management*] lookup. Follow these steps to add a new analysis property:

1. Add a new class that implements the `IHeaderPropertyHandler` interface. This interface contains a single `Check()` method that returns a value of `bool` type. Check the property value and return the result during the `Check()` method implementation. If the method returns `true`, the system creates the case using the standard mechanism, if `false`, the system treats the email as blacklisted.
2. Add an [Email header properties management] lookup value. Specify the property name used for analysis in the `Name` column. Specify the class name (added in the previous paragraph) in the `handler` column.

Working with email threads

The mechanism for creating email threads is available in Creatio since version 7.10.0. The main purpose of this mechanism is to improve the email interface. Use this mechanism to easily find email connections, e.g. by copying the connections of the previous email.

The thread construction mechanism is based on the `In-Reply-To` email header data. According to generally accepted agreements, this header should contain the `Message-ID` email identifier. The Creatio application retrieves these headers from the synchronized Email service (IMAP / Exchange).

The mechanism can be divided into two logical parts – creating threads and populating activity connection fields.

Creating email threads

The `EmailMessageData` detail contains 3 fields:

- `MessageId` – a string of 500 characters in length;
- `InReplyTo` – a string of 500 characters in length;
- `ParentMessageId` – a lookup field that references the `EmailMessageData` detail.

The `MessageId` and `InReplyTo` fields are populated with the corresponding message header values during

synchronization.

The `ParentMessageId` field is populated with the following values:

1. The `EmailMessageData` table searches for records where `MessageId` is identical to `InReplyTo`. The first found record is set as the current `ParentMessageId`.
2. The `ParentMessageId` field is populated with the current `Id` value in all `EmailMessageData` records if the `InReplyTo` field of these records is identical to the `MessageId` field.

Email thread connections are updated for every email.

Copying previous email connections in a thread

Spreading activity connections across the thread upon adding an email. The [`Case`] field is used in this case.

A parent record with the activity that includes the populated [`Contact`] field is searched for the current `EmailMessageData` record. The [`Case`] field values from this activity will be spread across the thread. Starting with the found `EmailMessageData` record, all child `EmailMessageData` records are found and their [`Case`] field value is updated.

A thread with 3 emails:

ActivityId	Title	CaseId	EmailMessageId	EmailMessageParentId
28BD6D59-B9D7-4FF9-89F5-FEE1DD003912	Re: relation	NULL	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3	F05B529D-C98C-4E26-BE00-21F8721AEF58
DC0A40D4-700A-40EB-B394-90E0376C3B5D	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	F05B529D-C98C-4E26-BE00-21F8721AEF58	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B
D7A9B82C-ED46-437C-980A-B2650D4FF3DA	relation	906909E8-4D64-47FD-AF92-B65B0826AEC3	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B	NULL

Another email is received in the thread:

ActivityId	Title	CaseId	EmailMessageId	EmailMessageParentId
6623B052-73AD-4AE5-AE61-A6F9BCD930A0	Re: relation	NULL	60C00B01-D0BF-40F6-923E-1830E433AEA1	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3
28BD6D59-B9D7-4FF9-89F5-FEE1DD003912	Re: relation	NULL	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3	F05B529D-C98C-4E26-BE00-21F8721AEF58
DC0A40D4-700A-40EB-B394-90E0376C3B5D	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	F05B529D-C98C-4E26-BE00-21F8721AEF58	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B
D7A9B82C-ED46-437C-980A-B2650D4FF3DA	relation	906909E8-4D64-47FD-AF92-B65B0826AEC3	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B	NULL

Starting with the record in which `EmailMessageId` is "60C00B01-D0BF-40F6-923E-1830E433AEA1", a record with the populated `CaseId` column is searched (does not contain `NULL`). This is a record where `EmailMessageId` = "F05B529D-C98C-4E26-BE00-21F8721AEF58", and `CaseId` = "1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B".

Now, starting with the record in which `EmailMessageId` is "F05B529D-C98C-4E26-BE00-21F8721AEF58", Creatio updates the value of the `CaseId` field for the connected records throughout the thread.

ActivityId	Title	Caseld	EmailMessageId	EmailMessageParentId
6623B052-73AD-4AE5-AE61-A6F9BCD930A0	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	60C00B01-D0BF-40F6-923E-1830E433AEA1	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3
28BD6D59-B9D7-4FF9-89F5-FEE1DD003912	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3	F05B529D-C98C-4E26-BE00-21F8721AEF58
DC0A40D4-700A-40EB-B394-90E0376C3B5D	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	F05B529D-C98C-4E26-BE00-21F8721AEF58	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B
D7A9B82C-ED46-437C-980A-B2650D4FF3DA	relation	906909E8-4D64-47FD-AF92-B65B0826AEC3	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B	NULL

Note.

Mail servers sometimes send out letters in the wrong sequence, disregarding the way they were written in the thread

(e.g. during synchronization, emails are received first from the inbox and then from the outbox). This complicates the mechanism. Building a thread for the emails that were not received consistently is impossible during synchronization. In that case, a thread can be built once the synchronization is complete. If certain mailbox folders are not loaded, or if the conversation was interrupted by other participants, the thread search logic may not work. However, the thread can be built when all emails are downloaded from the inbox in most cases.

Recommendations for adding a custom search process for all thread connections after downloading an email

Use the following guidelines to start working on a new email after thread formation:

1. The `Id` field of the synchronization session appears in the `EmailMessageData` table (the values are unique for all synchronization processes). This field is populated only for synchronized emails.
2. A record is added to the new `FinishedSyncSession` table if at least one email was received during synchronization.
3. Certain processes that responded to the signal of saving activities now respond to the (`Inserted`) insertion

signal of the `FinishedSyncSession` object. Emails from `EmailMessageData` are selected based on the synchronization session `Id`. The `MailboxSyncSettings` field of the `EmailMessageData` object can be used to select Email messages from the service box.

Macro handler in the email template

The email templates can be used for communication with a service team. They are available in the [*Email templates*] lookup. More information about email templates setup can be found in the “[Automatic emailing setup](#)”.

For example, the [*Case closure notification*] template is used to notify the user that the case has been closed. A number of pre-set macros can be used to display the values of some system object columns in the emails (for example contact's title or job).

Creatio enables to implement a custom logic of adding values that is returned by the macro handler. During the processing of the macro the Creatio will execute the algorithm implemented by the developer instead of the basic logic.

The `@Invoke` tag points on the specific handler of a class. After that, the name of the class that implements the `IMacrosInvokable` interface with the `GetMacrosValue()` method should be specified after “.”. This method will return the string that replaces the macro string.

To implement custom macro handler:

1. Create class that implements the `IMacrosInvokable` interface.
2. Register a macro in the `EmailTemplateMacros` table by specifying the `ParentId` (base template with the `@Invoke` tag) and the `ColumnPath` (class name).
3. Add a macro to the email template

Add a email filtering property

 Medium

Case description

Add a new `No-reply` property for case analysis. If the property is found in the email header and its value is anything other than “No”, treat the case as blacklisted.

Case implementation algorithm

1. Add a new class that implements the `IHeaderPropertyHandler` interface.

Add a “Source code” schema in a custom package (e.g. `Custom`). In this schema, define a class that implements the `IHeaderPropertyHandler` interface, e.g.:

```
namespace Terrasoft.Configuration
{
    using System;
```



```

public class NoreplyHandler: IHeaderPropertyHandler
{
    public bool Check(object value) {
        return string.Equals(value.ToString(), "No", StringComparison.OrdinalIgnoreCase);
    }
}

```

Save and publish the new schema.

The `IHeaderPropertyHandler` interface is defined in the `JunkFilter` package, so it must be added to the custom package dependency.

2. Adding a lookup value

Add the `No-reply` property to the [*Email header properties management*] lookup. Select the `NoreplyHandler` class (created in the previous paragraph) as the handler class (`handler` property).

After receiving an email with a `No-reply` header flag and a value that is anything other than “No”, the following options are possible:

- the case is not created if the [Create Cases From Junk Emails] system setting value is `false` ;
- the case is created with the [Junk Case Default Status] system setting status if the [Create Cases From Junk Emails] system setting value is `true` .

Add the macro handler to the email template



Advanced

The example is relevant to Service Creatio products.

Case description

Add the email macro handler that will return the “Test” string.

Source code

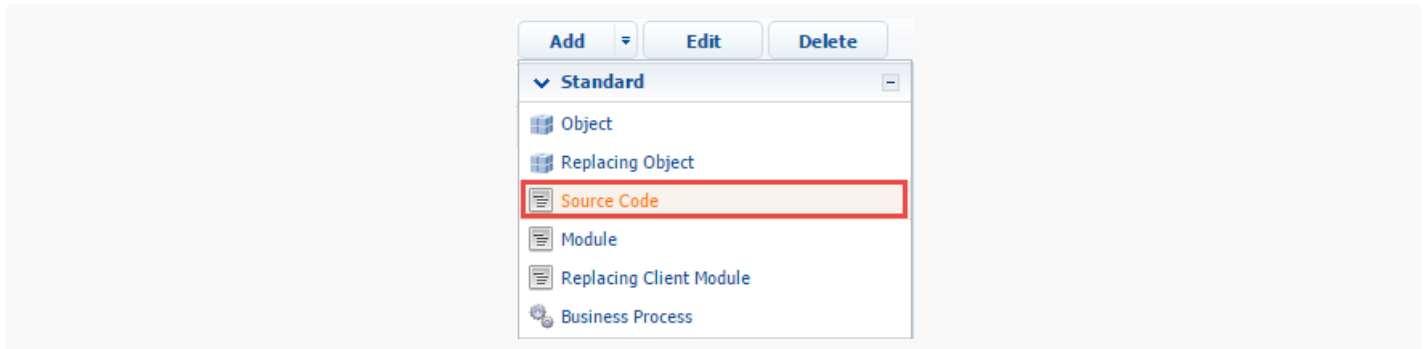
You can download the package with case implementation using the following [link](#).

Case implementation algorithm

1. Creating the class which implements the `IMacrosInvokable` interface.

To create class implementing the `IMacrosInvokable` interface, add the [*Source Code*] schema to the development package. For this, execute the [*Add*] - [*Source Code*] menu command on the [*Schemas*] tab in [the \[*Configuration* \]](#) section (Fig.1).

Fig. 1. Adding the [*Source Code*] schema



For the selected schema specify:

- [Title] - "Text string generator".
- [Name] - "UsrTestStringGenerator".

Source code of the schema:

```
namespace Terrasoft.Configuration
{
    using System;
    using Terrasoft.Core;
    // The class of the macro handler for the Email template.
    public class UsrTestStringGenerator : IMacrosInvokable
    {
        // A user connection.
        public UserConnection UserConnection {
            get;
            set;
        }
        // A method that returns a substitution value.
        public string GetMacrosValue(object arguments) {
            return "Test";
        }
    }
}
```

Publish the schema.

2. Macro registration in the EmailTemplateMacros table

To register macro in the `EmailTemplateMacros`, execute the following SQL request:

```

INSERT INTO EmailTemplateMacros(Name, Parentid, ColumnPath)
VALUES (
    'UsrTestStringGenerator',
    (SELECT TOP 1 Id
     FROM EmailTemplateMacros
     WHERE Name = '@Invoke'),
    'Terrasoft.Configuration.UsrTestStringGenerator'
)

```

3. Adding a macro to the email template

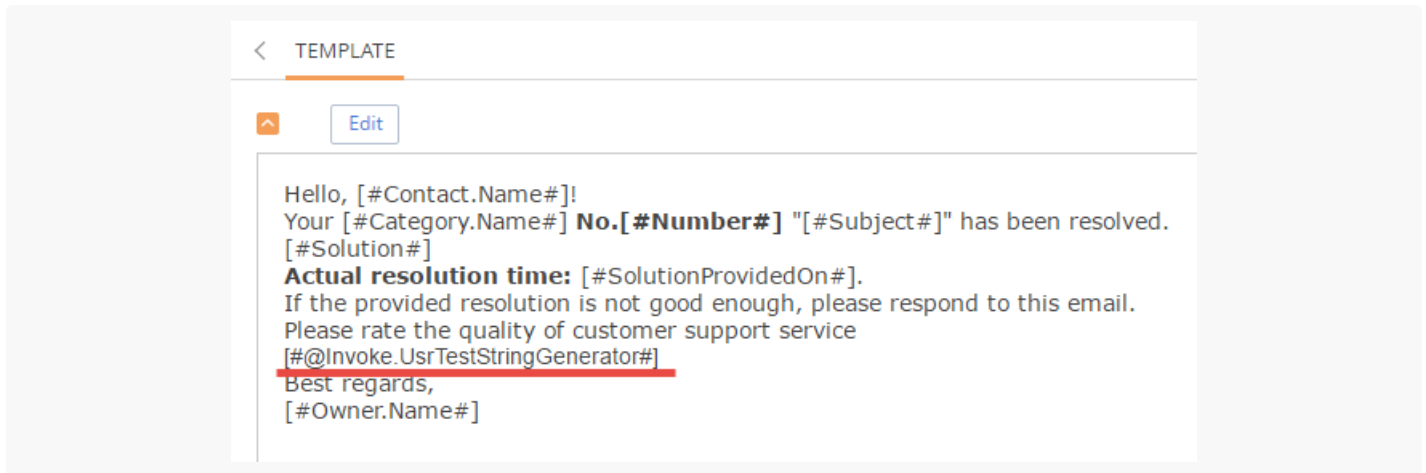
After registering the macro you can use it in the email templates. For this, modify one or several records of the [*Email templates*] lookup (Fig. 2).

Fig. 2. The [*Email templates*] lookup

The screenshot displays the 'Lookups' page in bpmonline. The 'Email templates' section is visible, listing several records. The record 'Case resolution notification' is highlighted with a red box. Its subject is 'Case #[Number] "[Subject]" is resolved'. Below the record are buttons for 'OPEN', 'COPY', and 'DELETE'. Other records include 'Case assigned to group', '"Offer of the day" template', and 'Template for new order approval notification'. The interface also shows a sidebar with navigation options and a top navigation bar with a search bar and user profile.

For example to modify the content of the emails sent at case resolution you need to change the [*Case resolution notification*] record. If you add the [*#@Invoke.UsrTestStringGenerator#*] macro to the template (Fig. 3), the "Test" value will be displayed instead of the macro in the email sent to the customer.

Fig. 3. Macro in the email template



Sending emails

 Advanced

Attention. Creatio version 7.16 and up supports sending emails using the explicit account credentials

In Creatio, you can send emails using both customer and developer means. By developer means, you can send emails:

- Using an existing account
- Using the explicit account credentials

Sending an email is done by using a business process. To set up a business process, use the [\[Auto-generated page \]](#) and [\[Script task \]](#) elements.

Sending an email from an existing account algorithm

To send an email from an existing account:

1. Create a business process. The process must include [\[Auto-generated page \]](#) and [\[Script task \]](#) required elements.
2. Create a config file for the email.
3. Add an attachment (optional).
4. Run the business process to send the email.

Creating a config file for the email

To create a configuration file of the email, use the `Terrasoft.Mail.Sender.EmailMessage` class. Populate the parameters below to ensure the validity of your email:

Creating a configuration file of the email

```

var message = new Terrasoft.Mail.Sender.EmailMessage {
    // Sender email address.
    From = "Sender@email.com",
    // Recipient email addresses.
    To = List<string>{ "first@recepient.co", "second@recepient.co"},
    // Copy optional
    Cc = List<string>{ "first@recepient.co", "second@recepient.co"},
    // Hidden copy optional
    Bcc = List<string>{ "first@recepient.co", "second@recepient.co"},
    // The subject of the email.
    Subject = "Message subject",
    // Email body.
    Body = "Body",
    // Priority, Terrasoft.Mail.Sender.EmailPriority enumeration values.
    Priority = Terrasoft.Mail.Sender.EmailPriority.Normal
};

```

Adding attachments optional

You can add attachments to your email. To do this, populate the [*Attachments*] field. Attachments are essentially a list of `Terrasoft.Mail.Sender.EmailAttachments` instances.

Adding attachments to email

```

// Creating an attachment.
var attachment = new Terrasoft.Mail.Sender.EmailAttachment {
    // Attachment ID.
    Id = new Guid("844F0837-EAA0-4F40-B965-71F5DB9EAE6E"),
    // Attachment name.
    Name = "attachName.txt",
    // Data.
    Data = byteData
};
// Adding the attachment to the message.
message.Attachments.Add(attachment);

```

Sending the email

To send the email, use the `Send` method of the `EmailSender` class with the email parameters and the connection configuration file.

Sending the email

```

// Sending the configured email message. To ignore access permissions when sending the message.
// set the ignoreRight parameter to "true".

```

```
emailSender.Send(message, ignoreRights);
```

Sending an email using the explicit account credentials algorithm

1. Create a business process. The process must include [*Auto-generated page*] and [*Script task*] required elements.
2. Create an instance of the `EmailClientFactory` class.
3. Create an instance of the `EmailSender` class.
4. Create a config file for connecting to the mailbox.
5. Create a config file of the email.
6. Add an attachment (if applicable).
7. Run the business process to send the email.

Creating an instance of the `EmailClientFactory` class

To create an instance of the `EmailClientFactory` class, make sure you have `UserConnection` established.

Creating an instance of the `EmailClientFactory` class

```
var emailClientFactory = ClassFactory.Get<IEmailClientFactory>(
    new ConstructorArgument("userConnection", UserConnection));
```

Creating an instance of the `EmailSender` class

To create an instance of the `EmailSender` class, pass the created `EmailClientFactory` instance and the `UserConnection` to the constructor.

Creating an instance of the `EmailSender` class

```
var emailSender = ClassFactory.Get<IEmailSender>(
    new ConstructorArgument("emailClientFactory", emailClientFactory),
    new ConstructorArgument("userConnection", UserConnection));
```

Creating a config file for connecting to the mailbox

To create a config file for connecting to the mailbox, use the `EmailContract.DTO.Credentials` class. Populate the following parameters:

Creating a config file for connecting to the mailbox

```

var credentialConfig = new EmailContract.DTO.Credentials {
    // Outbound mailserver IP address or domain
    ServiceUrl = "mail service host",
    // Can be left empty for some protocols
    Port = "Port",
    // Use SSL to encrypt the connection.
    UseSsl = false,
    // Mailbox user name
    UserName = "EmailUserName",
    // Mailbox user password
    Password = "UserPassword",
    // Mail server type (Exchange or IMAP/SMTP)
    ServerTypeId = EmailDomain.IntegrationConsts.ExchangeMailServerTypeId ||
        EmailDomain.IntegrationConsts.ImapMailServerTypeId,
    // Sender mailbox
    SenderEmailAddress = "sender@test.com"
};

```

Attention. The `ServiceUrl`, `UserName`, `Password`, `ServerTypeId`, `SenderMailbox` parameters are required.

Creating a config file of the email

To create a configuration file of the email that you are sending, use the `EmailContract.DTO.Email` class. Populate the parameters shown below to ensure the validity of your email:

Creating a config file of the email

```

var message = new EmailContract.DTO.Email {
    // Sender email address.
    Sender = "Sender@email.com",
    // Recipient email addresses.
    Receptions = List<string>{ "first@reception.co", "second@reception.co"},
    // The subject of the email.
    Subject = "Message subject",
    // Email body.
    Body = "Body",
    // Priority, EmailContract.EmailImportance enumeration values.
    Importance = EmailContract.EmailImportance.High
};

```

Adding attachments (if applicable)

You can add attachments to your email. To do this, populate the `[Attachments]` field. Attachments are essentially

a list of `EmailContract.DTO.Attachment` instances.

Adding attachments

```
// Creating an attachment.
var attachment = new EmailContract.DTO.Attachment {
    Name = "FileName",
    Id = "844F0837-EAA0-4F40-B965-71F5DB9EAE6E"
};
// Setting data for the attachment
attachment.SetData(byteData);
// Adding the attachment to the message.
message.Attachments.Add(attachment);
```

Sending the email

To send the email, use the `Send` method of the `EmailSender` class with the passed arguments of the email and the connection configuration file.

Sending the email

```
// Sending a configured email message with connection parameters
// for the mailbox of the sender To ignore access permissions when sending the message.
// set the ignoreRight parameter to "true".
emailSender.Send(message, credentialConfig, ignoreRights);
```

Send email from an existing account



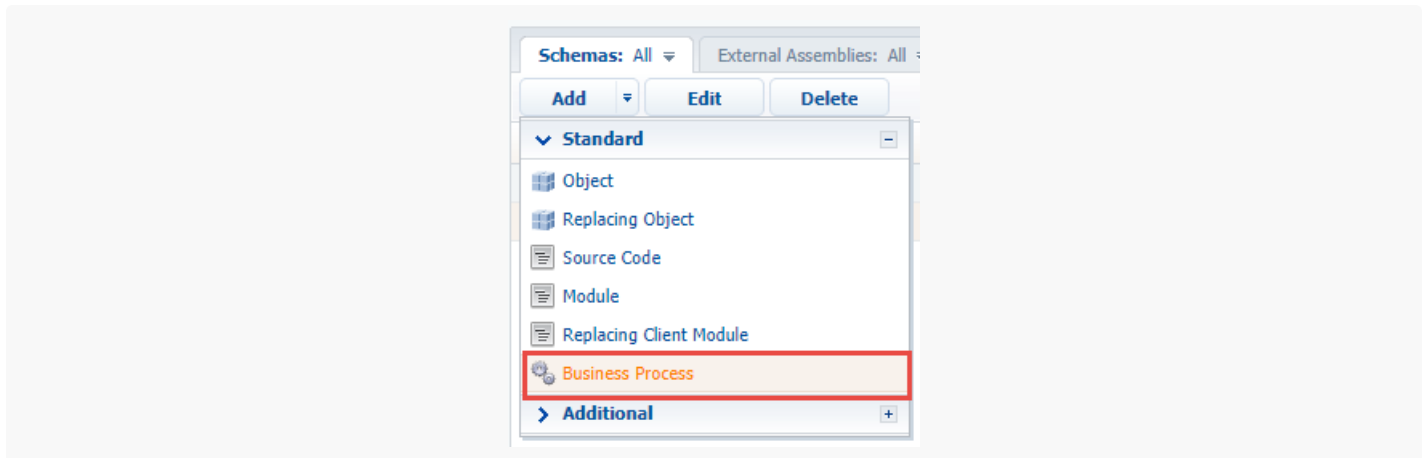
Advanced

Example. Create a business process that will open a page with email parameters fields to send an email using an existing account.

Example implementation algorithm

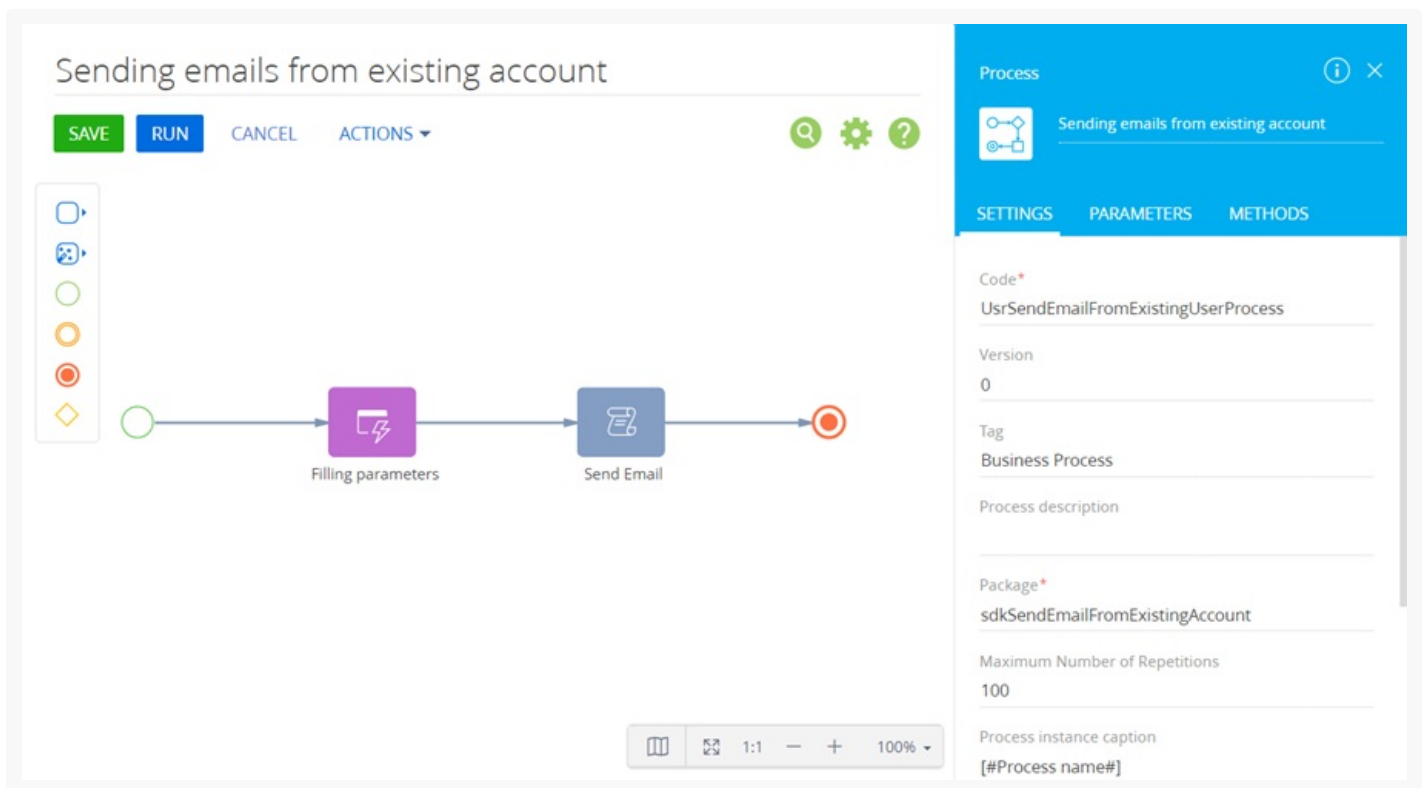
1. Create a business process

In the [*Configuration*] section execute the [*Add*] → [*Business process*] action.



In the opened Process Designer, set the following values for the properties in the setup area:

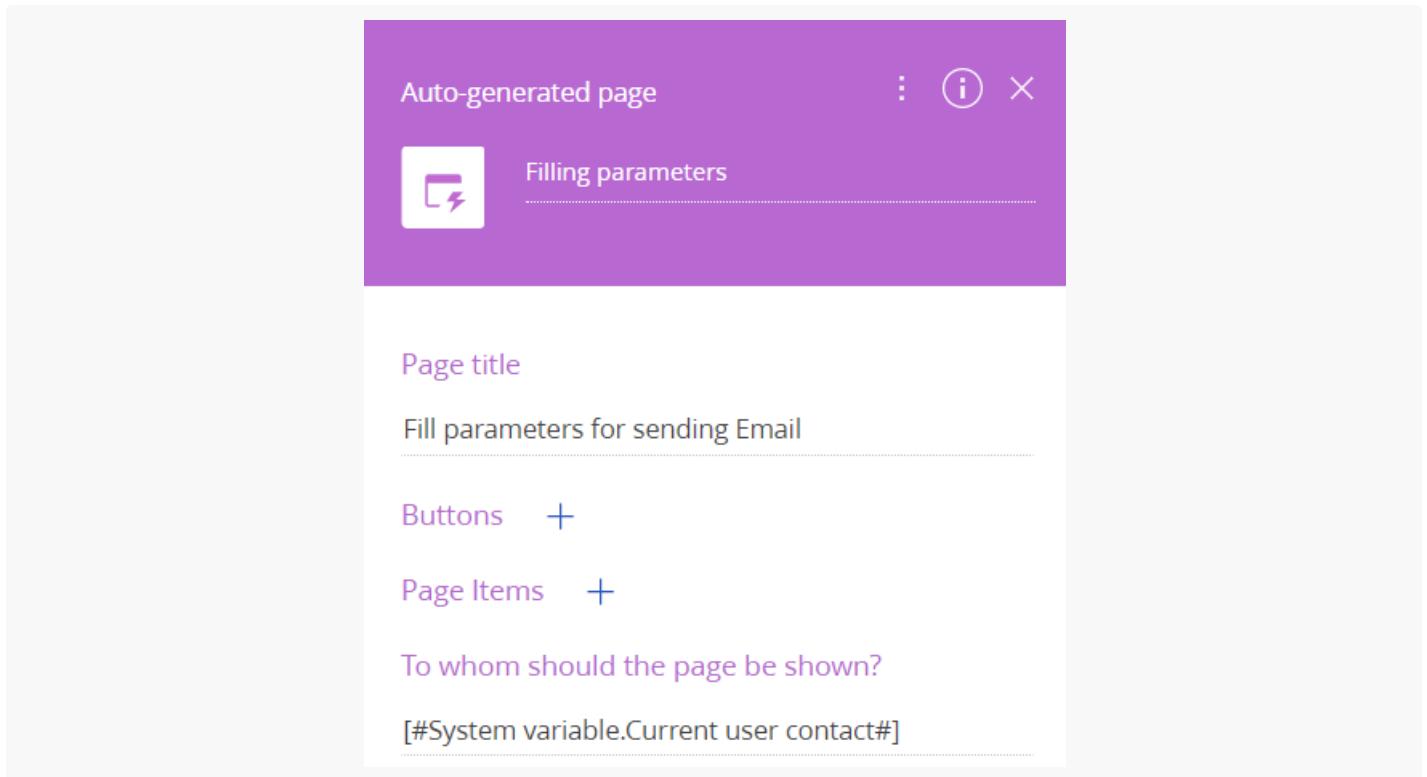
- [*Title*] — "Sending emails from existing account".
- [*Code*] — "UsrSendEmailFromExistingUserProcess".



2. Add the [*Auto-generated page*] element

The [*Auto-generated page*] element enables the process to open an arbitrary page created by the user. For this element, add "Filling parameters" as a caption and set the following properties:

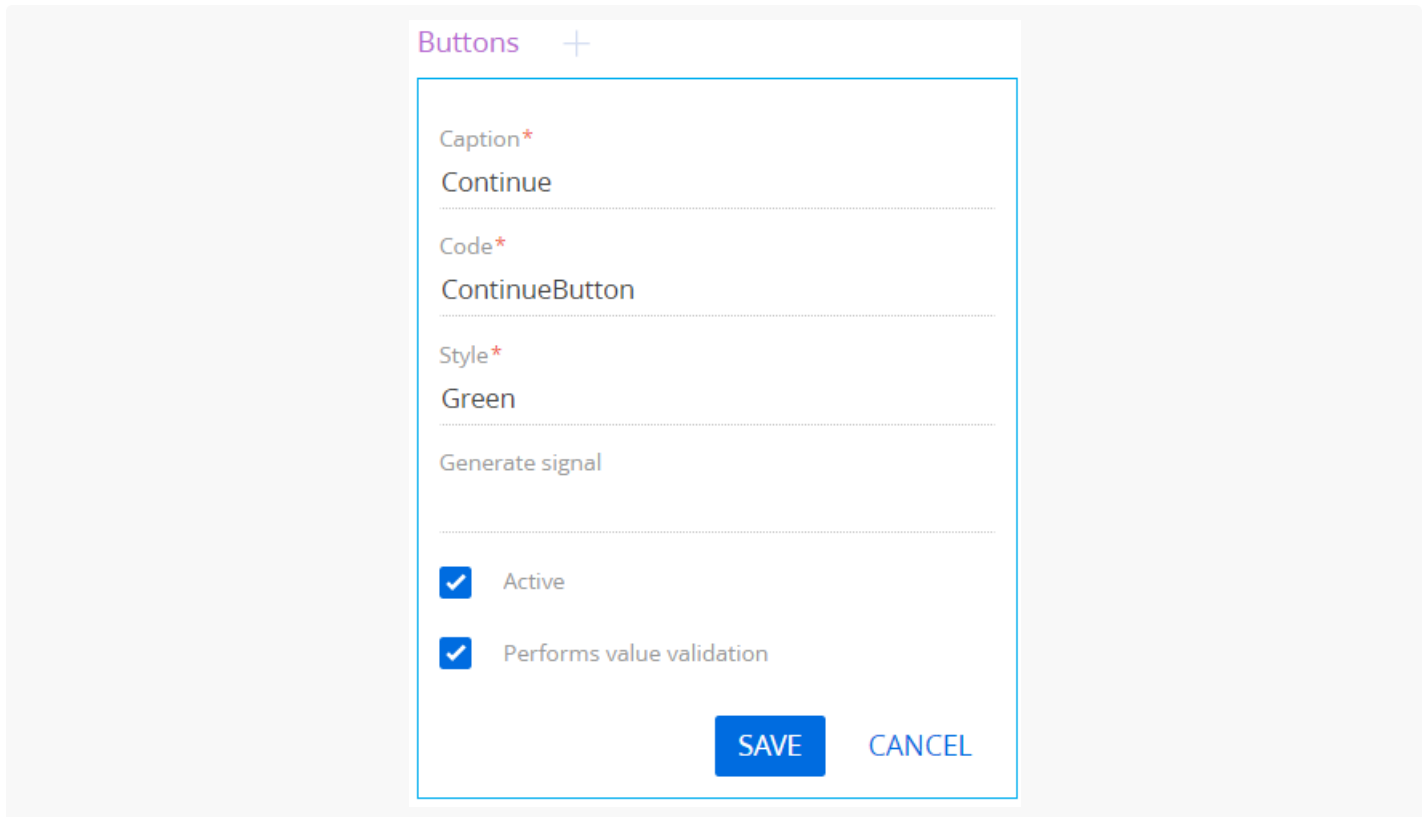
- [*Page title*] — "Fill parameters for sending Email".
- [*To whom should the page be shown?*] - select "Formula" and specify [*#System variable.Current user contact#*].



3. Add a button to the page

To add a [*Continue*] button in the [*Buttons*] block, click + and specify the following parameters:

- [*Caption*] — "Continue".
- [*Code*] - "ContinueButton".
- [*Style*] — select "Green".
- Set the [*Active*] checkbox.
- Set the [*Performs value validation*] checkbox.

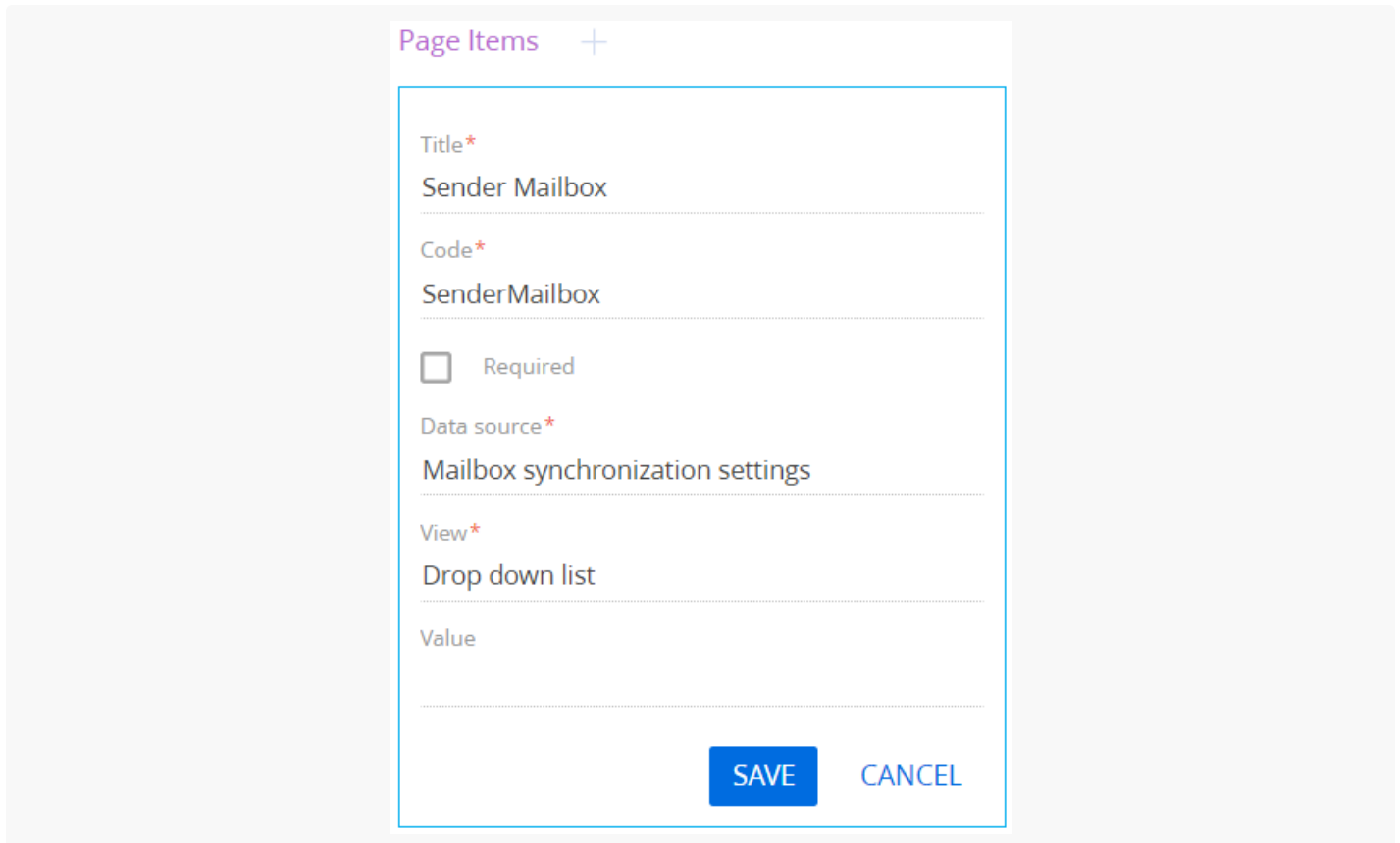


Click [Save].

4. Add elements to the page.

To add an element that will contain the email sender's mailbox, click **+** in the [*Page Items*] block, select the [*Selection field*] type and specify the following parameters:

- [*Title*] - "Sender Mailbox".
- [*Code*] - "SenderMailbox".
- [*Data source*] — select "Mailbox synchronization settings".
- [*View*] — select "Drop down list".



Page Items +

Title*
Sender Mailbox

Code*
SenderMailbox

Required

Data source*
Mailbox synchronization settings

View*
Drop down list

Value

SAVE CANCEL

Click [Save].

To add an element that will contain the email recipient's mailbox, click + in the [*Page Items*] block, select the [*Text field*] type and specify the following parameters:

- [*Title*] — "Recipient (many recipients separated by semicolon ";")".
- [*Code*] - "Recipient".
- Set the [*Required*] checkbox.

A configuration dialog box with a light blue border. It contains the following fields and options:

- Title***: A text input field with the placeholder text "Recipient (many recipients separated by se...".
- Code***: A text input field with the placeholder text "Recipient".
- Is multiline**: An unchecked checkbox.
- Required**: A checked checkbox.
- Value**: A text input field.

At the bottom right, there are two buttons: a blue **SAVE** button and a **CANCEL** button.

Click [Save].

To add an element that will contain the subject of the email, click + in the [Page Items] block, select the [Text field] type and specify the following parameters:

- [Title] - "Subject".
- [Code] - "Subject".
- Set the [Required] checkbox.

A configuration dialog box with a light blue border. It contains the following fields and options:

- Title***: A text input field with the placeholder text "Subject".
- Code***: A text input field with the placeholder text "Subject".
- Is multiline**: An unchecked checkbox.
- Required**: A checked checkbox.
- Value**: A text input field.

At the bottom right, there are two buttons: a blue **SAVE** button and a **CANCEL** button.

Click [Save].

To add an element that will contain the body of the email, click + in the [Page Items] block, select the [Text field] type and specify the following parameters:

- [*Title*] - "Body".
- [*Code*] - "Body".
- Set the [*Required*] checkbox.

Click [Save].

5. Add a [*Script task*] element

Set the value of the [*Title*] property of the [*Script task*] element to "Send Email" The element must execute the following program code:

Script Task code

```
var mailboxSettingId = Get<Guid>("SenderMailbox");
var emailClientFactory = ClassFactory.Get<IEmailClientFactory>(
    new ConstructorArgument("userConnection", UserConnection));
var emailSender = ClassFactory.Get<IEmailSender>(
    new ConstructorArgument("emailClientFactory", emailClientFactory),
    new ConstructorArgument("userConnection", UserConnection));

var entity = UserConnection.EntitySchemaManager.GetInstanceByName("MailboxSyncSettings").CreateEntity(
    if (entity.FetchFromDB("Id", mailboxSettingId, new List<string> { "SenderEmailAddress" }))) {
    var senderEmailAddress = entity.GetTypedColumnValue<string>("SenderEmailAddress");
    var message = new Terrasoft.Mail.Sender.EmailMessage {
        From = senderEmailAddress,
        To = Get<string>("Recipient").Split(';').ToList<string>(),
        // Cc = List<string>{ "first@receptient.co", "second@receptient.co"},
        // Bcc = List<string>{ "first@receptient.co", "second@receptient.co"},
        Subject = Get<string>("Subject"),
```

```

        Body = Get<string>("Body"),
        Priority = Terrasoft.Mail.Sender.EmailPriority.Normal
    };
    var attachment = new Terrasoft.Mail.Sender.EmailAttachment {
        Id = Guid.NewGuid(),
        Name = "test.txt",
        Data = Encoding.ASCII.GetBytes("some test text")
    };
    message.Attachments.Add(attachment);
    emailSender.Send(message);
}

return true;

```

6 Add parameters

To add a business process parameter that will contain the email recipient's mailbox, execute the [*Add parameters*] → [*Text*] action in the [*Parameters*] tab of the setup area and specify the following parameter properties:

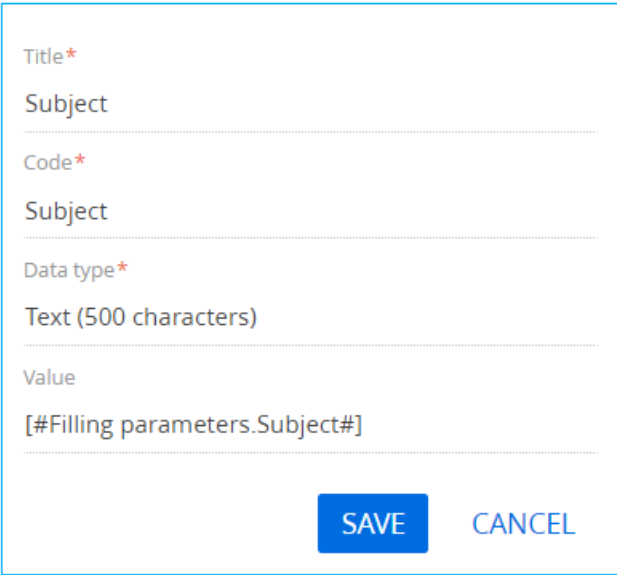
- [*Title*] - "Recipient".
- [*Code*] - "Recipient".
- [*Value*] - click ⚡ → [*Process parameter*] and select the "Recipient (many recipients separated by semicolon ";")" process element.

Click [*Save*].

To add a business process parameter that will contain the subject of the email, execute the [*Add parameters*] → [*Text*] action in the [*Parameters*] tab of the setup area and specify the following parameter properties:

- [*Title*] - "Subject".

- [*Code*] - “Subject”.
- [*Value*] - click ⚡ -> [*Process parameter*] and select the “Subject” process element.



A screenshot of a configuration dialog box for a parameter. The dialog has a light blue border and contains the following fields:

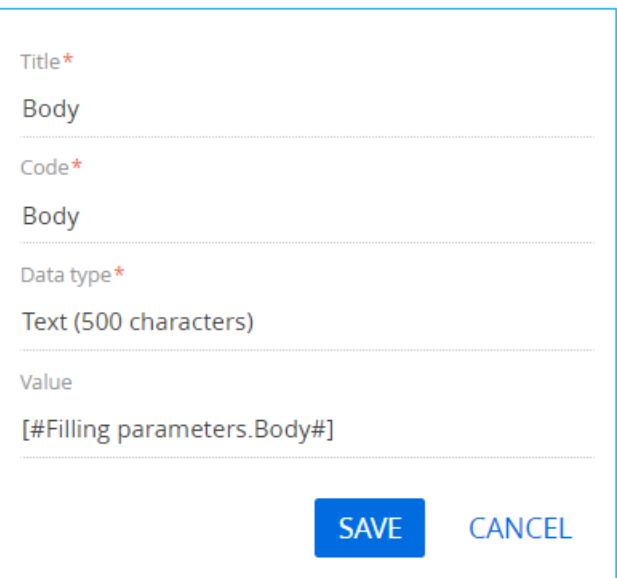
- Title***: Subject
- Code***: Subject
- Data type***: Text (500 characters)
- Value**: [#Filling parameters.Subject#]

At the bottom right, there are two buttons: a blue **SAVE** button and a **CANCEL** button.

Click [*Save*].

To add a business process parameter that will contain the body of the email, execute the [*Add parameters*] —> [*Text*] action in the [*Parameters*] tab of the setup area and specify the following parameter properties:

- [*Title*] - “Body”.
- [*Code*] - “Body”.
- [*Value*] - click ⚡ -> [*Process parameter*] and select the “Body” process element.



A screenshot of a configuration dialog box for a parameter. The dialog has a light blue border and contains the following fields:

- Title***: Body
- Code***: Body
- Data type***: Text (500 characters)
- Value**: [#Filling parameters.Body#]

At the bottom right, there are two buttons: a blue **SAVE** button and a **CANCEL** button.

Click [*Save*].

To add a business process parameter that will contain the email sender's mailbox, execute the [*Add parameters*] → [*Other*] → [*Unique identifier*] action in the [*Parameters*] tab of the setup area and specify the following parameter properties:

- [*Title*] - "Sender Mailbox".
- [*Code*] - "SenderMailbox".
- [*Value*] - click ⚡ → [*Process parameter*] and select the "Sender Mailbox" process element.

The screenshot shows a configuration dialog box with the following fields and values:

- Title***: Sender Mailbox
- Code***: SenderMailbox
- Data type***: Unique identifier
- Value**: [#Filling parameters.Sender Mailbox#]

At the bottom right of the dialog, there are two buttons: a blue **SAVE** button and a **CANCEL** button.

Click [*Save*].

7. Add methods

To add business process methods, click + in the [*Usings*] block in the [*Methods*] tab of the process designer setup area, and specify Terrasoft.Configuration value in the [*Name Space*] field. Click [*Save*].

Using the same method, add the following namespaces:

- Terrasoft.Mail.Sender
- Terrasoft.Core.Factories
- Terrasoft.Core
- Terrasoft.Mail
- IntegrationApi
- System.Linq

Save all changes in the Process Designer.

8. Run the business process

Attention. To run the business process successfully, make sure that a user account for the email sender is available in the Creatio application. Learn more about adding a user account in the “[Integration with the MS Exchange service](#)” block of articles.

After the business process is run using the [*Run*] button, a page containing fields for specifying email parameters will open.

Fill parameters for sending Email

CONTINUE CLOSE

Sender Mailbox

Recipient (many recipients separated by semicolon ";")*

Subject*

Body*

To send an email from the corresponding email account, click [*Continue*].

Send email using the explicit account credentials

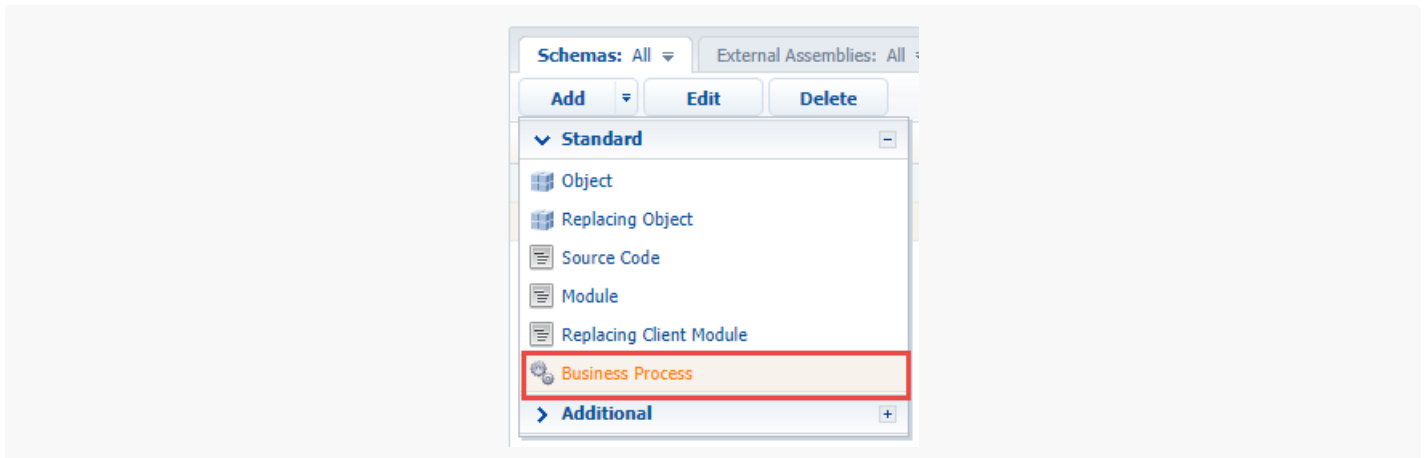
 **Advanced**

Example. Create a business process that will open a page containing fields mapped to the email parameters to send an email using the explicit account credentials.

Example implementation algorithm

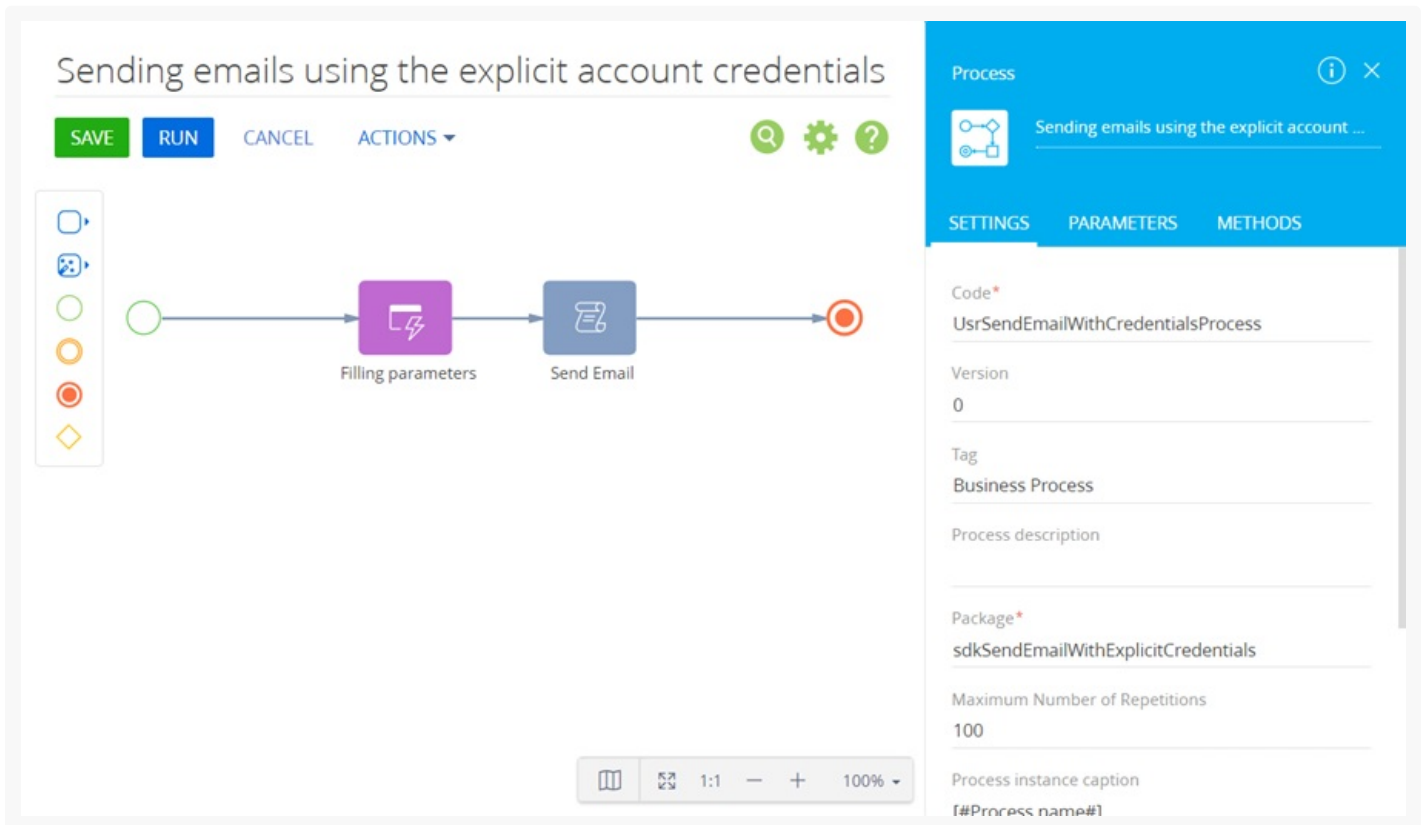
1. Create a business process

In the [*Configuration*] section execute the [*Add*] → [*Business process*] action.



In the opened process designer set the following values for the properties in the setup area:

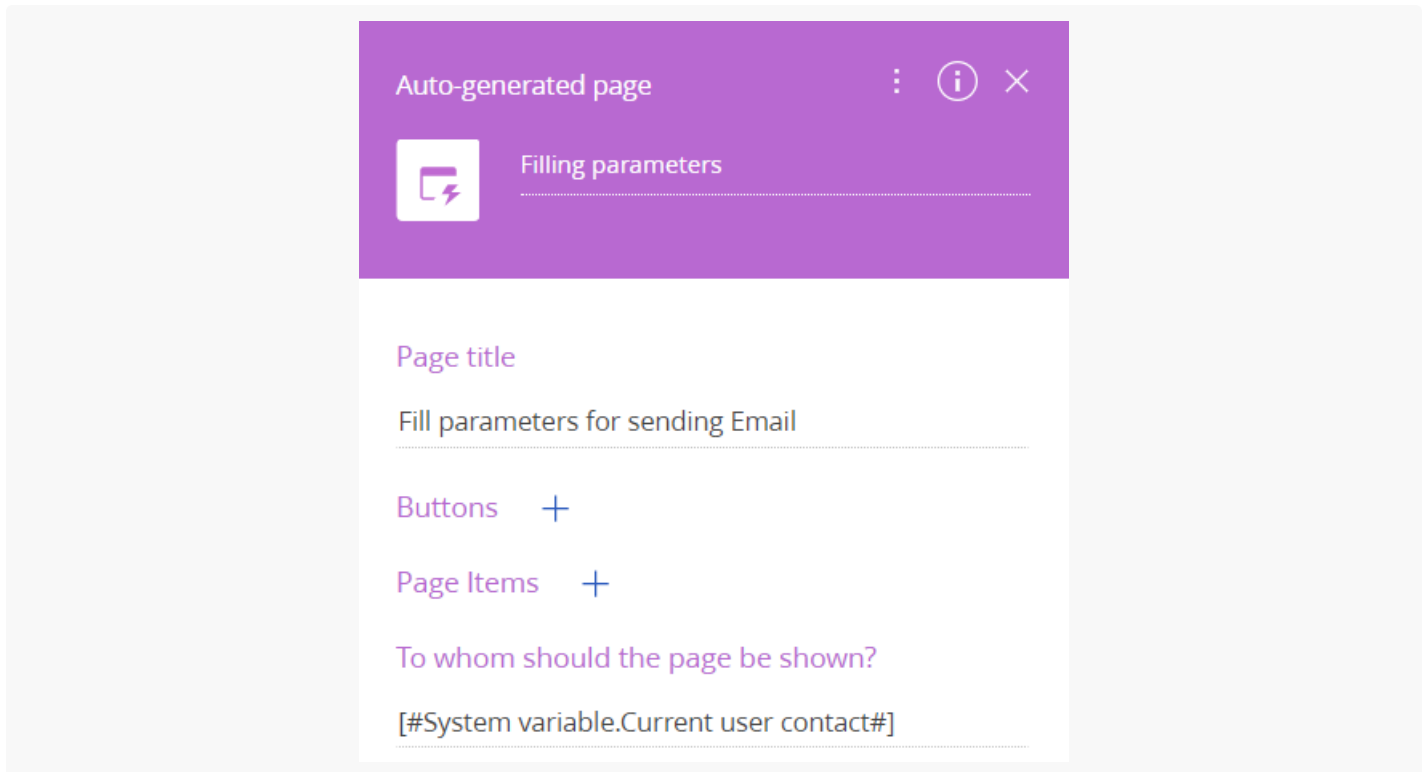
- [*Title*] — "Sending emails using the explicit account credential".
- [*Code*] - "UsrSendEmailWithCredentialsProcess".



2. Add the [*Auto-generated page*] element

The [*Auto-generated page*] element enables the process to open an arbitrary page created by the user. For this element, add [*Filling parameters*] as a caption and set the following properties:

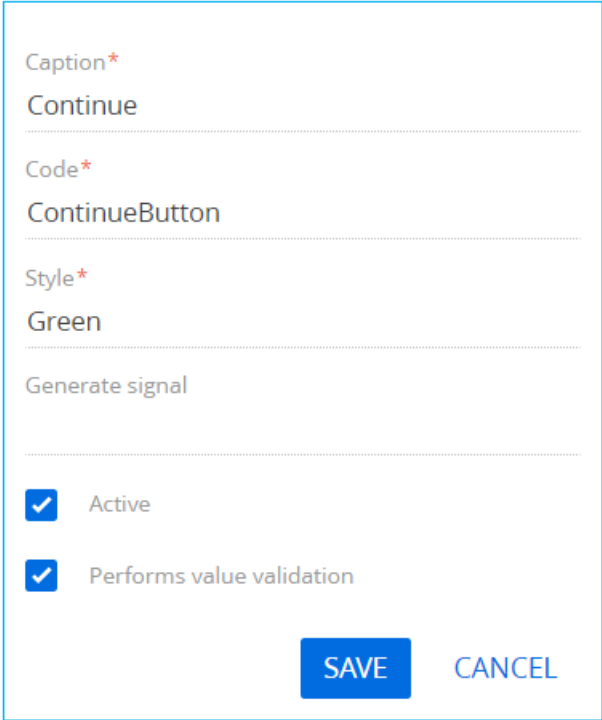
- [*Page title*] — "Fill parameters for sending Email".
- [*To whom should the page be shown?*] - select "Formula" and specify [*#System variable.Current user contact#*].



3. Add a button to the page

To add a [*Continue*] button, click + in the [*Buttons*] block and specify the following parameters:

- [*Caption*] — "Continue".
- [*Code*] - "ContinueButton".
- [*Style*] — select "Green".
- Set the [*Active*] checkbox.
- Set the [*Performs value validation*] checkbox.



Configuration dialog for a button:

- Field: Caption*
Value: Continue
- Field: Code*
Value: ContinueButton
- Field: Style*
Value: Green
- Field: Generate signal
Value:
- Field: Active
Value:
- Field: Performs value validation
Value:

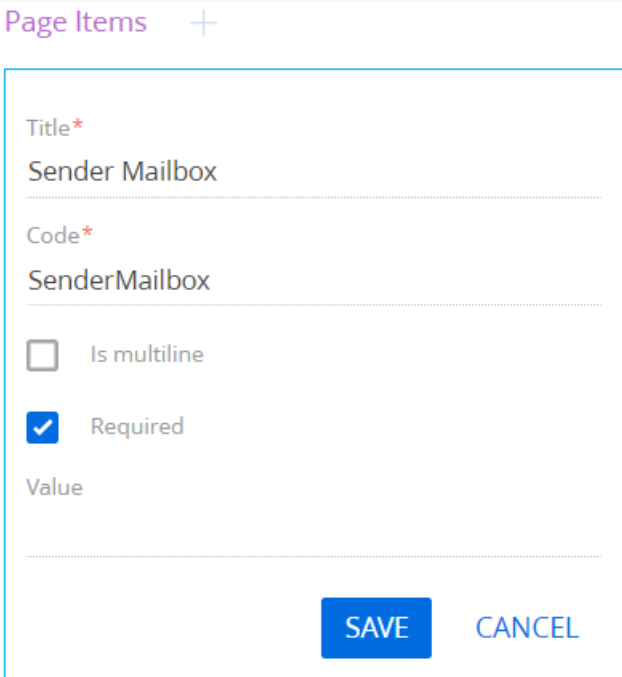
Buttons: SAVE, CANCEL

Click [Save].

4. Add elements to the page.

To add an element that will contain the email sender's mailbox, click **+** in the [*Page Items*] block, select the [*Text field*] type and specify the following parameters:

- [*Title*] - "Sender Mailbox".
- [*Code*] - "SenderMailbox".
- Set the [*Required*] checkbox.



Page Items +

Title*
Sender Mailbox

Code*
SenderMailbox

Is multiline

Required

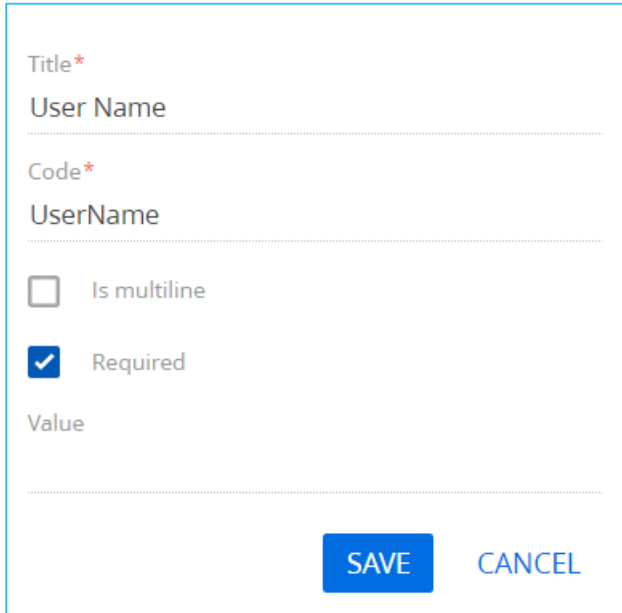
Value

SAVE CANCEL

Click [Save].

To add an element that will contain the name of the email sender, click + in the [Page Items] block, select the [Text field] type and specify the following parameters:

- [Title] - "User Name".
- [Code] - "UserName".
- Set the [Required] checkbox.



Title*
User Name

Code*
UserName

Is multiline

Required

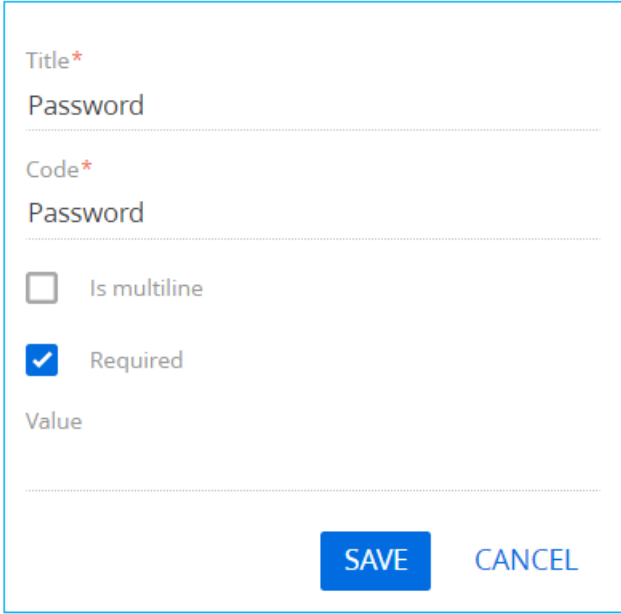
Value

SAVE CANCEL

Click [Save].

To add an element that will contain the password for email sender's mailbox, click **+** in the [*Page Items*] block, select the [*Text field*] type and specify the following parameters:

- [*Title*] - "Password".
- [*Code*] - "Password".
- Set the [*Required*] checkbox.



The screenshot shows a configuration dialog for a text field. The dialog has a light gray background and a white content area. It contains two text input fields, both with "Password" entered. The first field is labeled "Title*" and the second is labeled "Code*". Below the fields are two checkboxes: "Is multiline" (unchecked) and "Required" (checked). At the bottom, there is a "Value" label and a dotted line. At the bottom right, there are two buttons: "SAVE" (blue) and "CANCEL" (light blue).

Click [*Save*].

To add an element that will contain the mail server address of the email sender, click **+** in the [*Page Items*] block, select the [*Text field*] type and specify the following parameters:

- [*Title*] - "Service Url".
- [*Code*] - "ServiceUrl".
- Set the [*Required*] checkbox.

A configuration dialog box with a light gray background and a white content area. The dialog has a blue border. It contains the following fields and options:

- Title***: A text input field containing "Service Url".
- Code***: A text input field containing "ServiceUrl".
- Is multiline**: A checkbox that is unchecked.
- Required**: A checkbox that is checked.
- Value**: A text input field that is empty.

At the bottom right of the dialog are two buttons: a blue "SAVE" button and a gray "CANCEL" button.

Click [Save].

To add an element that will contain the port number of the email sender's mail provider, click **+** in the [*Page Items*] block, select the [*Integer*] type and specify the following parameters:

- [*Title*] - "Port".
- [*Code*] - "Port".

A configuration dialog box with a light gray background and a white content area. The dialog has a blue border. It contains the following fields and options:

- Title***: A text input field containing "Port".
- Code***: A text input field containing "Port".
- Required**: A checkbox that is unchecked.
- Value**: A text input field that is empty.

At the bottom right of the dialog are two buttons: a blue "SAVE" button and a gray "CANCEL" button.

Click [Save].

To add an element that will contain the cryptographic protocol to ensure a secure connection, click **+** in the [*Page Items*] block, select the [*Boolean*] type and specify the following parameters:

- [*Title*] - " Use Ssl".
- [*Code*] - "UseSsl".

Title*
 Use Ssl
 Code*
 UseSsl
 Value
 SAVE CANCEL

Click [Save].

To add an element that will contain the email recipient's mailbox, click + in the [*Page Items*] block, select the [*Text field*] type and specify the following parameters:

- [*Title*] — "Recipient (many recipients separated by semicolon ";")".
- [*Code*] - "Recipient".
- Set the [*Required*] checkbox.

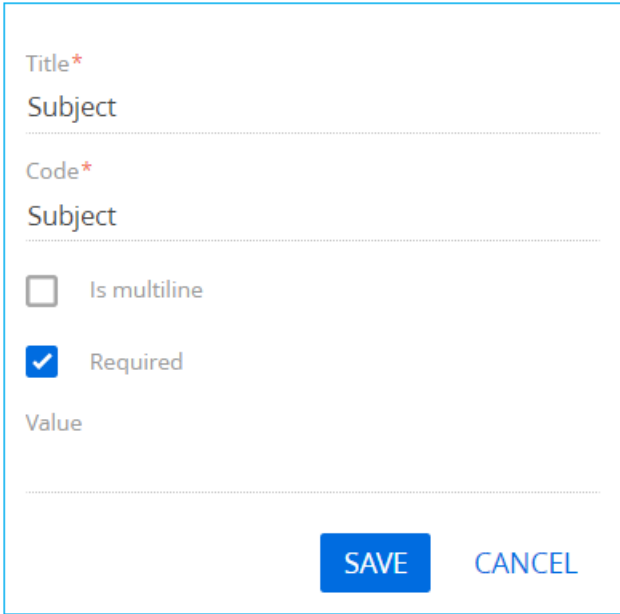
Title*
 Recipient (many recipients separated by se...
 Code*
 Recipient
 Is multiline
 Required
 Value
 SAVE CANCEL

Click [Save].

To add an element that will contain the subject of the email, click + in the [*Page Items*] block, select the [*Text field*] type and specify the following parameters:

- [*Title*] - "Subject".
- [*Code*] - "Subject".

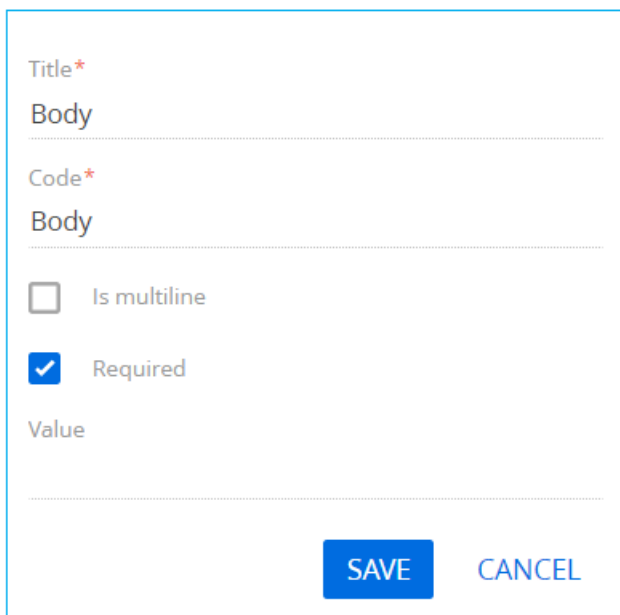
- Set the [*Required*] checkbox.



Click [*Save*].

To add an element that will contain the body of the email, click **+** in the [*Page Items*] block, select the [*Text field*] type and specify the following parameters:

- [*Title*] - “Body”.
- [*Code*] - “Body”.
- Set the [*Required*] checkbox.



Click [*Save*].

To add an element that will contain the type of the email sender's provider, click **+** in the [*Page Items*] block, select the [*Selection field*] type and specify the following parameters:

- [*Title*] — "Type of mail server".
- [*Code*] - "ServerTypeId".
- Set the [*Required*] checkbox.
- [*Data source*] — select "Mail service provider type".
- [*View*] — select "Drop down list".

Click [*Save*].

5. Add a [*ScriptTask*] element

Set the value of the [*Title*] property of the [*Script task*] element to "Send Email". The element must execute the following program code:

Script Task code

```
var emailClientFactory = ClassFactory.Get<EmailClientFactory>(new ConstructorArgument("userConne
var credentialConfig = new EmailContract.DTO.Credentials {
    ServiceUrl = Get<string>("ServiceUrl"),
    Port = Get<int>("Port"),
    UseSsl = Get<bool>("UseSsl"),
    UserName = Get<string>("UserName"),
    Password = Get<string>("Password"),
```

```

    ServerTypeId = Get<Guid>("ServerTypeId"),
    SenderEmailAddress = Get<string>("SenderMailbox")
};
var emailSender = ClassFactory.Get<IEmailSender>(new ConstructorArgument("emailClientFactory", e
    new ConstructorArgument("userConnection", UserConnection));
var message = new EmailContract.DTO.Email {
    Sender = credentialConfig.SenderEmailAddress,
    Receipients = Get<string>("Recipient").Split(';').ToList<string>(),
    Subject = Get<string>("Subject"),
    Body = Get<string>("Body"),
    Importance = EmailContract.EmailImportance.Normal,
    IsHtmlBody = true,
    // CopyReceipients = new List<string> { "user@mail.service" },
    // BlindCopyReceipients = new List<string> { "user@mail.service" }
};
var attachment = new EmailContract.DTO.Attachment {
    Id = Guid.NewGuid().ToString(),
    Name = "test.txt",
};
byte[] data = Encoding.ASCII.GetBytes("some test text");
attachment.SetData(data);
message.Attachments.Add(attachment);
emailSender.Send(message, credentialConfig);
return true;

```

6 Add parameters

To add a business process parameter that will contain the address of the sender's mail server, execute the [*Add parameters*] → [*Text*] action in the [*Parameters*] tab of the setup area and specify the following parameter properties:

- [*Title*] - "Service Url".
- [*Code*] - "ServiceUrl".
- [*Value*] - click ⚡ → [*Process parameter*] and select the "Service Url" process element.

Title*

Service Url

Code*

ServiceUrl

Data type*

Text (500 characters)

Value

[#Filling parameters.Service Url#]

SAVE CANCEL

Click [Save].

To add a business process parameter that will contain the port number of the sender's email provider, execute the [Add parameters] → [Integer] action in the [Parameters] tab of the setup area and specify the following parameter properties:

- [Title] - "Port".
- [Code] - "Port".
- [Value] - click ⚡ → [Process parameter] and select the "Port" process element.

Title*

Port

Code*

Port

Data type*

Integer

Value

[#Filling parameters.Port#]

SAVE CANCEL

Click [Save].

To add a business process parameter that will contain the cryptographic protocol for a secure connection, execute the [Add parameters] → [Boolean] action in the [Parameters] tab of the setup area and specify the following parameter properties:

- [*Title*] - " Use Ssl".
- [*Code*] - " UseSsl".
- [*Value*] - click ⚡ -> [*Process parameter*] and select the " Use SSL" process element.

The screenshot shows a dialog box with the following fields and values:

- Title***: Use Ssl
- Code***: UseSsl
- Data type***: Boolean
- Value**: [#Filling parameters.Use Ssl#]

At the bottom right, there are two buttons: **SAVE** and **CANCEL**.

Click [*Save*].

To add a business process parameter that will contain the email sender's name, execute the [*Add parameters*] -> [*Text*] action in the [*Parameters*] tab of the setup area and specify the following parameter properties:

- [*Title*] - "User Name".
- [*Code*] - "UserName".
- [*Data type*] - select "Text (250 characters)".
- [*Value*] - click ⚡ -> [*Process parameter*] and select the "User Name" process element.

The screenshot shows a dialog box with the following fields and values:

- Title***: User Name
- Code***: UserName
- Data type***: Text (250 characters)
- Value**: [#Filling parameters.User Name#]

At the bottom right, there are two buttons: **SAVE** and **CANCEL**.

Click [Save].

To add a business process parameter that will contain the password for the email sender's mailbox, execute the [Add parameters] → [Text] action in the [Parameters] tab of the setup area and specify the following parameter properties:

- [Title] - "Password".
- [Code] - "Password".
- [Data type] - select "Text (250 characters)".
- [Value] - click ⚡ → [Process parameter] and select the "Password" process element.

The screenshot shows a configuration dialog for a parameter. The fields are as follows:

- Title***: Password
- Code***: Password
- Data type***: Text (250 characters)
- Value**: [#Filling parameters.Password#]

At the bottom right, there are two buttons: **SAVE** (in a blue box) and **CANCEL** (in a grey box).

Click [Save].

To add a business process parameter that will contain the email recipient's mailbox, execute the [Add parameters] → [Text] action in the [Parameters] tab of the setup area and specify the following parameter properties:

- [Title] - "Recipient".
- [Code] - "Recipient".
- [Value] - click ⚡ → [Process parameter] and select the "Recipient (many recipients separated by semicolon ";")" process element.

Title*

Recipient

Code*

Recipient

Data type*

Text (500 characters)

Value

[#Filling parameters.Recipient (many recipi...]

SAVE CANCEL

Click [Save].

To add a business process parameter that will contain the type of the email sender's mail provider, execute the [Add parameters] → [Other] → [Unique identifier] action in the [Parameters] tab of the setup area and specify the following parameter properties:

- [Title] – "Type of mail server".
- [Code] – "ServerTypeld".
- [Value] – click ⚡ → [Process parameter] and select the "Type of mail server" process element.

Title*

Type of mail server

Code*

ServerTypeld

Data type*

Unique identifier

Value

[#Filling parameters.Type of mail server#]

SAVE CANCEL

Click [Save].

To add a business process parameter that will contain the email sender's mailbox, execute the [Add parameters] → [Text] action in the [Parameters] tab of the setup area and specify the following parameter properties:

- [*Title*] - "Sender Mailbox".
- [*Code*] - "SenderMailbox".
- [*Data type*] - select "Text (250 characters)".
- [*Value*] - click ⚡ -> [*Process parameter*] and select the "Sender Mailbox" process element.

The image shows a configuration dialog box with the following fields and values:

- Title***: Sender Mailbox
- Code***: SendeMailbox
- Data type***: Text (250 characters)
- Value**: [#Filling parameters.Sender Mailbox#]

At the bottom right of the dialog are two buttons: **SAVE** and **CANCEL**.

Click [Save].

7. Add methods

To add business process methods, click + in the [*Usings*] block in the [*Methods*] tab of the process designer setup area, and specify `Terrasoft.Mail.Sender` value in the [*Name Space*] field. Click [Save].

Using the same method, add the following namespaces:

- `Terrasoft.Core.Factories`
- `System.Linq`

Save all changes in the Process Designer.

8. Business process launch

After the business process is run using the [*Run*] button, a page containing fields for specifying email parameters will open.

Fill parameters for sending Email

[CONTINUE](#)[CLOSE](#)

Sender Mailbox*

User Name*

Password*

Service Url*

Port 0

Use Ssl

Recipient (many recipients separated by semicolon ";")*

Subject*

Body*

Type of mail server*

To send an email using the explicit account credentials, click [*Continue*].