KSL FOR SALESFORCE®

PROJECT CONFIGURATION GUIDE

May 2020

naelan

Sommaire

Terms of use of this guide

The rights to use the software described herein are assigned under a license agreement and this guide may only be used or copied in accordance with the terms of the contract.

The information in this guide is subject to change without notice.

The reproduction or transmission of the information in this guide is limited to internal use by the customer and for the sole purpose of proper use of the software. Any other reproduction or transmission is prohibited without the express written permission of NAELAN.

This guide is provided by NAELAN for information on the software delivered. It does not in any way constitute a contractual commitment both on the features indicated and in their implementation.

Unless otherwise stated, the companies, names and data used in our examples are fictitious ; any reconciliation with real companies or entities would be the result of coincidence.

Typographic conventions

Text file sample	Code example or file settings
ℹ	Note, information
📖 Chapter, Guide	Reference to another guide or chapter
Example	Example, variable, code extract
Keyword, label	Keyword, application label, important item

Contact Naelan

Headquarter - 4 rue Claude Chappe
69370 Saint-Didier au Mont d'Or France
+33 4 37 59 81 40

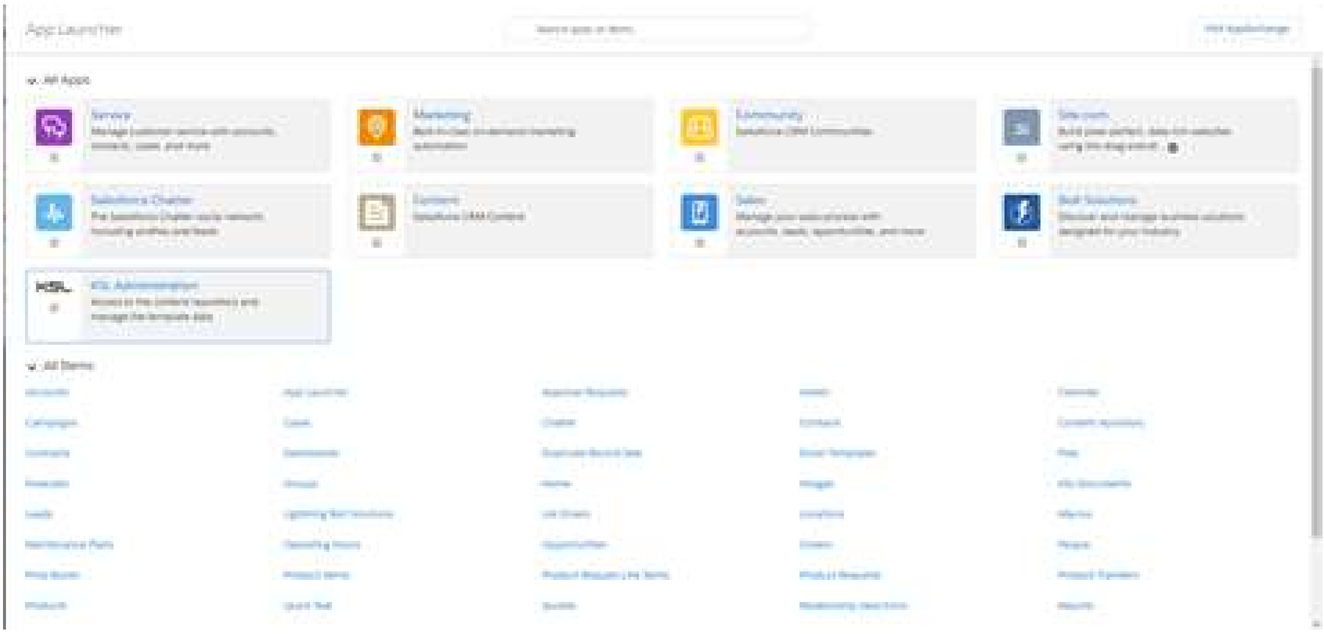
www.naelan.com
support@naelan.com

Office in Paris - 4 Place Louis Armand
75023 Paris France
+33 1 72 76 80 67

contact@naelan.com

1. KSL For Salesforce® for administrators

Select the application KSL Administration from the Salesforce® App Launcher menu.



App Launcher

All Apps

Service

Marketing

Community

Site.com

Salesforce Chatter

Content

Sales

Bolt Solutions

KSL Administration

All Items

This KSL Administration application only concerns KSL for Salesforce® plugin et does not response to the same needs as KSL Admin application,which is used to setup and supervise a KSL Server.

KSL Administration includes 2 tabs:

- Setup
- Content repository

1.1. Setup tab

The Setup tab contains 2 sub-tabs:

- Auth. Configuration
- XSD Generator

a. XSD Generator

This tab allows the administrator to generate an XSD schema associated with Salesforce® objects. It also allows you to generate a test data file to initiate the design of a document or e-mail template.

XSD use

This XSD is used by the KSL Studio design tool to build the document templates based on the data of the selected objects. A template is associated with an XSD. The same XSD can be used by several templates.

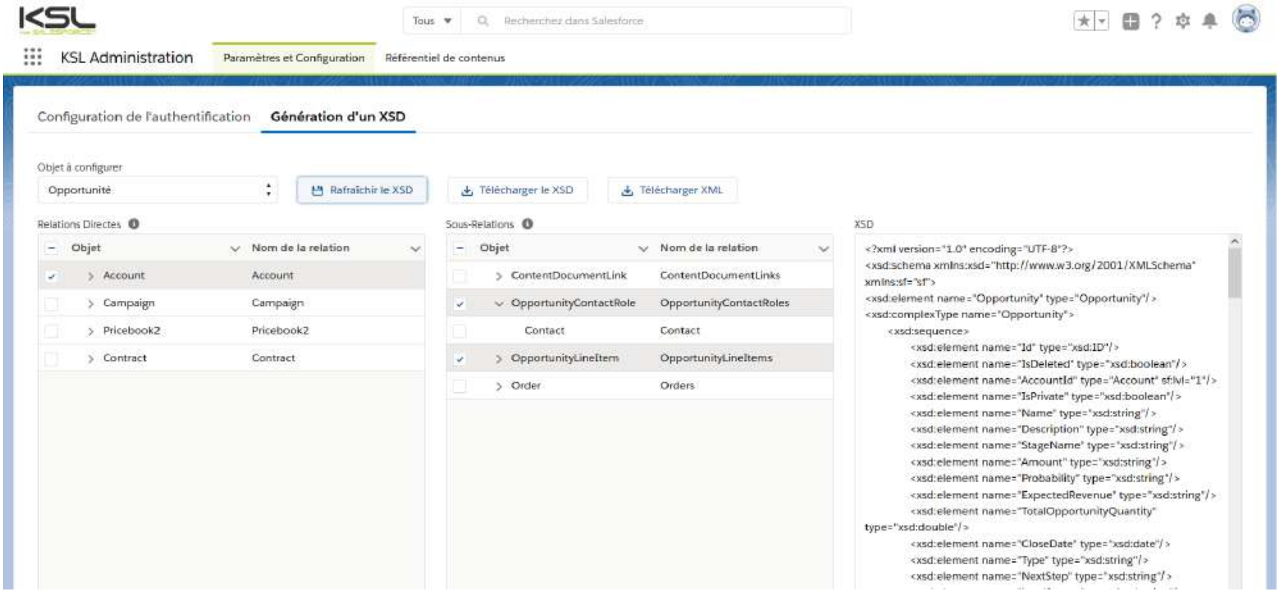
When generating a document, the KSL plug-in asks the KSL server for the XSD schema associated with the selected template; the plug-in then retrieves the necessary data in the Salesforce® objects, and instantly creates a stream in XML format with this data. This XML conforms to the XSD and allows the KSL server to provide the field values to generate the document.

Note: this exchange makes it possible to detect as soon as possible any discrepancies between the data available in Salesforce® and the data expected by the document template; without this mechanism, deleting for example a Salesforce® field could cause a problem when generating the document, or even generate a document with erroneous contents. In the event of such an offset, an error is displayed in Salesforce® when the document is generated.

XSD generation

The XSD generator allows you to select the main object of your XSD. The builder automatically proposes the direct relationship (Parent) and indirect relationship (Child) objects that exist for this object in your Salesforce® environment. The checkboxes allow you to choose the objects you want to include in your XSD, in relation to the fields you will use in your KSL document templates.

The objects available in the menu are configured in the Metadata Types, Allowed Object (more information in the chapter 3.2.c)



Objet à configurer: Opportunité

Relations Directes

	Objet	Nom de la relation
✓	› Account	Account
□	› Campaign	Campaign
□	› Pricebook2	Pricebook2
□	› Contract	Contract

Sous-Relations

	Objet	Nom de la relation
□	› ContentDocumentLink	ContentDocumentLinks
✓	⌄ OpportunityContactRole	OpportunityContactRoles
□	Contact	Contact
✓	› OpportunityLineItem	OpportunityLineItems
□	› Order	Orders

XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:sf="sf">
<xsd:element name="Opportunity" type="Opportunity"/>
<xsd:complexType name="Opportunity">
    <xsd:sequence>
        <xsd:element name="Id" type="xsd:ID"/>
        <xsd:element name="IsDeleted" type="xsd:boolean"/>
        <xsd:element name="AccountId" type="Account" sf:lvl="1"/>
        <xsd:element name="IsPrivate" type="xsd:boolean"/>
        <xsd:element name="Name" type="xsd:string"/>
        <xsd:element name="Description" type="xsd:string"/>
        <xsd:element name="StageName" type="xsd:string"/>
        <xsd:element name="Amount" type="xsd:string"/>
        <xsd:element name="Probability" type="xsd:string"/>
        <xsd:element name="ExpectedRevenue" type="xsd:string"/>
        <xsd:element name="TotalOpportunityQuantity"
type="xsd:double"/>
        <xsd:element name="CloseDate" type="xsd:date"/>
        <xsd:element name="Type" type="xsd:string"/>
        <xsd:element name="NextStep" type="xsd:string"/>
```

Note: You cannot select the same types of objects more than once in the same relationship tree. For example, if you select Account, you will not be able to select the parent Account because they belong to the same type of object in the same tree; KSL document templates will not allow it.

The Generation tab of an XSD includes several buttons for generating XSD and XML:

- The Refresh XSD button

Click this button after selecting the objects you want to insert into your XSD in order for the builder to generate XSD file in the right window. This window contains all the fields of the selected objects.

- The Download XSD button

This button allows downloading the XSD file and using it in KSL Studio to create a document template. It will not work if you have not previously clicked Refresh XSD.

When you create a document template based on this XSD, this XSD is linked to the template. With each KSL document generation (from a template), Salesforce® ask for this XSD and use it to retrieve the values of the Salesforce® object fields described in the schema.

Special attention when changing of the data schema:

- If you create a new field in an object used in a KSL document template:
  - If you want to use this field in your document, generate a new XSD, download it, update the XSD component in the KSL repository template and update the KSL template using it.
  - If you do not want to use the field, do not do anything. Users will still have the option of generating or customizing the KSL document in Salesforce® (the new field will be ignored).
  - Note that a new field created in your Saleforce® object must not have the prefix "ksl" which is a reserved prefix.

- If you delete a field of an object used in a KSL document template:
  - Generate and download a new XSD; download it, update the XSD component in the KSL repository template and update the KSL template using it.
  - The field should also be removed from the document layout. If you do not perform this update, an error will be returned to the users who will no longer be able to use the document template in Salesforce®.

- If some objects present in your XSD are not finally associated with the main object (opportunity for example)
  - The XML generated will not be consistent with the XSD, which will cause a document generation error.
  - However, it is possible to make these objects optional by manually modifying the XSD (minOccurs = O). Note that fields from potentially missing objects will be empty.

Note that it is possible to delete some fields of the XSD that would not be necessary for the template in order to reduce the size of the data stream exchanged between Salesforce® and the KSL servers, and also to limit the number of fields accessible from the KSL Studio tool to simplify the selection of fields during template design.

- The Download XML button (version 1.6 and higher of the KSL plug-in)

This button is used to generate and download a test data file compliant with the XSD schema. It will not work if you have not previously clicked Refresh XSD.

Each data is surrounded by XML tags:

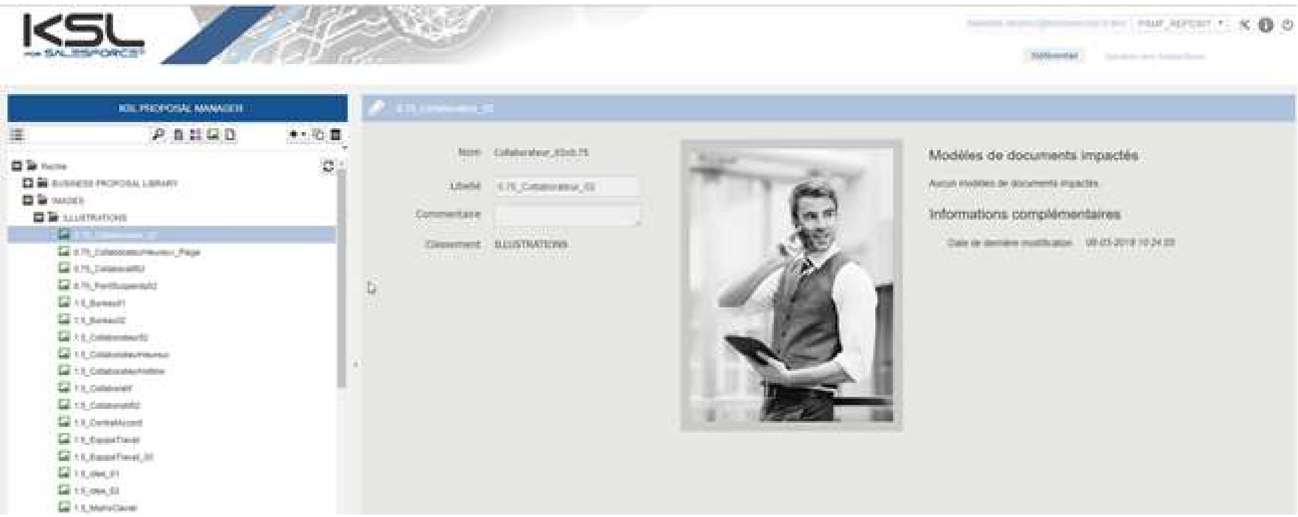
- A text field data is the field name
- A date field data is a random date
- An integer field data is a random integer
- A decimal field data is a random decimal number
- A boolean field data is set to "true" or "false"

Examples:

- <BillingStreet>BillingStreet</BillingStreet>
- <LastActivityDate>2020-09-12 00:00:00</LastActivityDa
- <NumberOfWonOpportunities>83</NumberOfWonOpportunities>
- <Quantity>58.04</Quantity>
- <IsDeleted>true</IsDeleted>

1.2. Content repository tab

This tab opens in KSL Office on a new tab, the repository containing shared components.
The functions of this application are detailed in KSL Office user guide.



Nom [illegible]

Libellé [illegible]

Commentaire

Classement ILLUSTRATIONS

Modèles de documents impactés

Aucun modèles de documents impactés

Informations complémentaires

Date de dernière modification [illegible]

2. KSL for Salesforce® for template designers

2.1. Mandatory KSL functional parameters

When creating new KSL templates, i.e. graphical editions of documents and the associated editing services, and e-mail templates, it is necessary to create the following 3 functional parameters of string type:

- P_FILE: This parameter is used to identify the XML stream transmitted by the plug-in to the editing service. It must be initialized to 250 characters
- P_LABEL: this parameter is used to transmit the description of the document; it corresponds to the field Document description - see chapter 5.1. and 5.2; it can be initialized to 50 characters
- language: This field passes the language used by the user in their Salesforce® application; this field is for example "FR" or "en_US"; we recommend setting it to 10 characters.

2.2. Other KSL functional parameters

Other KSL functional parameters can be added for use in the Document parameters view of the interactive document or email (for example, to allow the user to select a table of contents type or to select optional annexes).

These other parameters can not be valuated by Salesforce® through parameters; to meet this need, additional data will have to be added to the transmitted XML stream.

2.3. Considerations on Salesforce® data sent to KSL

Document and email generation with KSL relies on XML data conformed to an XSD schema.

Examples of XSD schemas and XML streams can be generated using the KSL Administration application. On this point, KSL Administration is only a practical tool that facilitates the work of the KSL template designer. See chapter KSL For Salesforce® for Administrators for more information.

2.4. Use of XML data in KSL

Before creating templates, be sure to set some of the following Advanced run parameters options in KSL Studio:

- XML file decimal character
- XML file thousands separator
- Date format in XML file

See the KSL Studio User Guide for more details.

This setting must also be reported in the project on KSL Server. Please contact the Naelan teams for this operation.

2.5. Create on-demand email template

To create on-demand e-mail templates, proceed as follows. This process is explained using the object Opportunity but any other object can be used.

Note: The following chapter is dedicated to the particular case of e-mailings.

a. Step 1: Generate the XSD and XML streams corresponding to an e-mail

In Ksl Administration, generate and download the XSD schema corresponding to the data personalizing the e-mail (and if necessary an associated XML test data file). This XSD schema will be used by the KSL service(s) in charge of creating the HTML bodies of e-mails.

Example: suppose that the body of the emails to be generated has to include the data corresponding to the Opportunity name and amount, but also lastname, the firstname of the first Contact Roles of the opportunity.

In this context, it is necessary to generate the XSD and XML files from the Opportunity object which contains the Name and Amount elements, but also the elements LastName, FirstName and Salutation of the OpportunityContactRole node of the opportunity.

Example of XML data corresponding to this XSD schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<Opportunity>
  <Name>My opportunity</Name>
  <Amount>10000,00</Amount>
  ...
  <OpportunityContactRole>
    <ContactId>
        <LastName>MARTIN</LastName>
        <FirstName>John</FirstName>
        <Salutation>Mr</Salutation>
    ...
```

b. Step 2: create a mapping file

The mapping file is used to create variables inserted in the body of an e-mail:

- These variables can be selected and inserted by the designer or the content administrator in the body of each e-mail template within text zones.

- These variables can be selected and inserted by the Salesforce user when he edits the body of his e-mail
- These variables are automatically substituted during the e-mail body generation with data from the XML stream coming from Salesforce

- Create a mapping file with a .map extension:
  - The mapping file is a text file
  - It defines each field used by an e-mail (field used in the subject, the body of the e-mail, or the recipient fields if it is an e-mailing)
  - Each field is defined by a line of the file

Each line is composed of 3 strings separated by the character ";"

- The 1st string is the name of the field that will be used in the e-mail: if used in the template, this field will be the one displayed in the interactive personalization interface, otherwise the end user will be able to insert it in a text zone of the e-mail in this interface
- The 2nd string is the element of the XML stream that contains the value to display (single-child node)
- The 3rd string is the element of the XML, which is the of the element defined just above, that contains the value to display

Example with the <SF_Email_Mapping.map> mapping file:

```
OPPORTUNITY_NAME;Name;Opportunity;
OPPORTUNITY_AMOUNT;Amount;Opportunity;
CONTACT_LASTNAME;LastName;ContactId;
CONTACT_FIRSTNAME;FirstName;ContactId;
CONTACT_SALUTATION;Salutation;ContactId;
```

Example:

The XML stream and the mapping file given above will allow you to automatically substitute the e-mail variables with the Salesforce data with the following result:

- *OPPORTUNITY_NAME = My opportunity*
- *OPPORTUNITY_AMOUNT = 10000,00*
- *CONTACT_LASTNAME = MARTIN*
- *CONTACT_FIRSTNAME = John*
- *CONTACT_SALUTATION = Mr*

Important notes:

- *Note that only the parent node of each element is to be defined*
- *It is possible to define several fields that point to the same element*
- *If one of the elements is iterative, the field takes the value of the element of the first iteration*
- *Each line ends with a line break (including the last line)*

  - *In the case where the parent element exists several times in the flow, it is possible to specify the parent node of the parent node (example: SALUTATION_CONTACT; Salutation; OpportunityContactRole.ContactId;)*

- From the KSL Administration application, classify this mapping file in the KSL repository:
- Select the Content repository tab
  - Add the mapping file as an external file and classify it in the tree structure

c. Step 3: Create a style sheet for your new e-mail template

If you already have a style sheet, go directly to the next step.

Otherwise, from KSL Studio (or KSL Office from KSL Server version 8.0.2.0), you can duplicate the FS_SALESFORCE_EMAIL style sheet and modify it to match your graphic charter.

d. Step 4: Create the body of your e-mail

From the Content repository tab of the KSL Administration application, create a group of text zones. This group will be the body of your e-mail template.

This group of text zones contains one or more text zones that are also created from the Content Repository application. These text zones must use the style sheet created in the previous step 3.

The variables correspond to the Salesforce data are created in the text zone that require it.

- The value of each variable comes from the XML stream sent by Salesforce
- Each variable is in string format
- The name of each variable corresponds to the 1st element of the line of the mapping file that defines it

For the example described above, the following variables will be created:

- *OPPORTUNITY_NAME*
- *OPPORTUNITY_AMOUNT*
- *CONTACT_LASTNAME*
- *CONTACT_FIRSTNAME*
- *CONTACT_SALUTATION*

e. Step 5: Create a new KSL document template for the e-mail template

In this context of creating a new e-mail template, it is necessary to create a new KSL document template from which a new KSL Server service will be created in step 6.

The procedure from KSL Studio is as follows:

- Add the XSD file generated in step 1 to the KSL project classification plan

- Duplicate the generic edition SF_Email_500px and add it to the KSL project classification tree
- In this new document template, select the style sheet by selecting the style sheet component created in step 3 above
- Modify the XML request for this document template by specifying the new XSD
- Delete the group in the document template
- Insert the new text zone group created in step 4
- Save and Update and release the document template

f. Step 6: Create the document service corresponding to the email template

- From KSL Admin, create the document service associated with the document template created in step 5
- Set a label in the description field: this description corresponds to the label displayed for this e-mail template in the list of templates presented to the Salesforce user in its Document Templates component.
- Classify this service in the project tree: the document service should be classified in a branch of the tree accessible to the Salesforce user in its Document Templates component accessible from the Opportunity object.
- Specify the preview action group: EmailActions
- Fill in the parameters of this document service using the information below

Parameter	Default value or example	Description
feuillestyles (stylesheet)	FS_SALESFORCE_EMAIL	Select the style sheet used for editing (without extension) Deselect the option "Show this parameter in Ksl Office"
MailAccount		*Parameter not used in the case of a single on demand e-mail sent by Salesforce®* Deselect the option "Show this parameter in Ksl Office"
MailBat	False	Set this parameter to False for this single on-demand e-mail sending case Deselect the option "Show this parameter in Ksl Office"
MailCharset	UTF-8	Set this parameter to UTF-8 *Encoding of the CSV file generated from the XML stream sent by the KSL plug-in* *Set this parameter by default to the universal UTF-8 code page; it is possible to change this encoding (example: UTF-8, Windows-1252, ISO8859-15 (Linux)* Deselect the option "Show this parameter in Ksl Office"
MailContentGroup	Example: <SF_Email_ContractConfirm.ksg>	*Optional parameter for transmitting the group of text fields used as e-mail body*

		If not specified, the group inserted by default in the document template is used *This parameter requires that the document template dynamically calls the group (default group dynamically called with the expression MailContentGroup)* *The group is defined between the characters < and > with the extension .ksg* Deselect the option "Show this parameter in Ksl Office"
MailCsvFile	<default>	This file is used only if the P_MAP_FILE file is not defined. The default value can be used for debugging tests without XML stream. A value between the characters < and > indicates that the file is in the KSL repository. The CSV file is a text file with one contact / recipient per line, possibly with a header line that names the columns and column separators ";" You need to set a default value (for example "<default>") even if the P_MAP_FILE is defined Deselect the option "Show this parameter in Ksl Office"
MailCsvHeader	Correspond to the header line of the CSV file without #	*Parameter not used in the case of a single on demand e-mail sent by Salesforce®* *This CSV is only used if P_MAP_FILE is not defined.* Deselect the option "Show this parameter in Ksl Office"
MailLinkFile	<emailing_links.csv>	Set if necessary the file giving the definition of hypertext links in the e-mail (see documentation on the KSL Email Designer function) Deselect the option "Show this parameter in Ksl Office"
MailReturnMode	1	Set this parameter to the value 1. Deselect the option "Show this parameter in Ksl Office" *For information:* - *1: for the creation of an e-mail sent by Salesforce* - *2: for the creation of an e-mailing sent by KSL Server or asynchronously sent by an external e-mail send & track provider*
MailTitle	default	*Parameter not used in the case of a single on demand e-mail sent by Salesforce®* This parameter must have a value (for example "default") Deselect the option "Show this parameter in Ksl Office"
MailUrl	Example: https://www.naelan.fr/ksl_images/	Set the web directory Url used for storing images (see the KSL Email Designer documentation) Deselect the option "Show this parameter in Ksl Office"

P_FILE	Example: <SF_email_Opportunity.xml>	*XML data stream passed by the KSL for Salesforce plug-in* *This default value is used only during debugging tests* Deselect the option "Show this parameter in Ksl Office"
P_LABEL		*Parameter not used in the case of a single on demand e-mail sent by Salesforce®* Deselect the option "Show this parameter in Ksl Office"
language	Example: FR	Salesforce interface language passed through the KSL plug-in Deselect the option "Show this parameter in Ksl Office"
P_MAP_FILE	Example: <SF_email_mapping.map>	Indicate the mapping file created for the template and corresponding to its XSD Deselect the option "Show this parameter in Ksl Office"

g. Step 7 (optional): create other e-mail templates based on the same data

In many business contexts, it is necessary to create a family of e-mail templates that share the same graphic design and based on the same Salesforce objects and data.

In this case, it may be more efficient to create KSL e-mail services based on the same KSL e-mail template. The procedure is then the following.

- In KSL Studio:
  - Edit the document template created in step 5
  - Reserve the template and edit the properties of the text zone group by right-clicking / Properties
  - In the window that appears, specify the expression MailContentGroup the Expression field of the dynamic group call
  - Validate, save and release the edition

- From the Content repository tab of the KSL Administration application,
  - Create as many groups of text boxes as desired body of e-mail (1 group per model of e-mail)

- In KSL Admin:
  - Recompile the document template above
  - Copy the document service as many times as needed (as much service as the family's email template)
  - Give a description to each copied service with the name of the template you want in Salesforce
  - Edit the functional parameters of each copied service and set the MailContentGroup parameter specifying the name of the group to call (Service 1> Group 1, Service 2> Group 2, etc.)

Benefits:

Thanks to this setting:

- All e-mail templates share the same document template, the same document service, and the same style sheet
- Creating and editing an e-mail body is done directly from Salesforce
- Adding a template is done in seconds by duplicating a service

Example:

SF_Email_500px service is an example of service shared by the 3 e-mail templates implemented by the following 3 services:

- SF_Email_ContractConfirm_500px
- SF_Email_ContractTermin_500px
- SF_Email_TrainingConfirm_500px

2.6. Create emailing templates

To create e-mailing templates, proceed as follows. This process is explained using the object Campaign but any other object can be used.

Note: The previous chapter is dedicated to the particular case of on-demand single e-mail.

a. Step 1b: Generate the XSD and XML streams corresponding to the e-mailing

In Ksl Administration, generate and download the XSD schema corresponding to the data personalizing the e-mailing (and if necessary an associated XML test data file). This XSD schema will be used by the KSL service(s) in charge of creating the HTML bodies of the e-mails.

Example:

- *Suppose that the body of the emails to be generated has to include the data corresponding to the fields Salutation, LastName and FirstName of the targeted contacts and the Campaign name.*
- *Moreover, the E-mail address of each targeted contact must also be used by the e- mailing, since the KSL server is responsible for sending, or transmitting e-mails generated to an external send & track provider.*

In this context, it is necessary to generate the XSD and XML files from the Campaign object which contains the Description element, but also from the CampaignMember sub-object which contains the elements LastName, FirstName, Salutation and Email.

Example of XML data corresponding to this XSD schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<Campaign>
  <Description>My emailing</Description>
  …
  <CampaignMember>
        <LastName>MARTIN</LastName>
        <FirstName>John</FirstName>
        <Salutation>Mr</Salutation>
        <Email>martin@naelan.com</Email>
        …
  </CampaignMember>
  <CampaignMember>
        <LastName>SIMONS</LastName>
        <FirstName>Marie</FirstName>
        <Salutation>Mrs</Salutation>
        <Email>simons@naelan.com</Email>
        …
  </CampaignMember>
  …
```

b. Step 2b: create a mapping file

The mapping file is used to create variables inserted in the body of an e-mailing:

- These variables can be selected and inserted by the designer or the content administrator in the body of each e-mailing template within text zones.
- These variables can be selected and inserted by the Salesforce user when he edits the body of the e-mailing
- These variables are automatically substituted during the e-mail bodies generation with data from the XML stream coming from Salesforce

- In KSL Admin, create a mapping file with a .map extension:
  - The mapping file is a text file
  - It defines each field used by an e-mail (field used in the subject, the body of the e-mail, or the recipient fields if it is an e-mailing)
  - Each field is defined by a line of the file

Each line is composed of 3 strings separated by the character ";"

- *The 1st string is the name of the field that will be used in the e-mail: if used in the template, this field will be the one displayed in the interactive personalization interface, otherwise the end user will be able to insert it in a text zone of the e-mail in this interface*
- *The 2nd string is the element of the XML stream that contains the value to display (single-child node)*
- *The 3rd string is the element of the XML, which is the of the element defined just above, that contains*

the value to display

Example with the <SF_Email_Mapping.map> mapping file:

```
CAMPAGNE_NAME;Description;Campaign;
CONTACT_LASTNAME;LastName;CampaignMember;
CONTACT_FIRSTNAME;FirstName;CampaignMember;
CONTACT_SALUTATION;Salutation;CampaignMember;
Email;Email;CampaignMember;
```

Example:
The XML stream and the mapping file given above will allow you to automatically substitute the e-mail variables with the Salesforce data with the following result:

For the 1st e-mail:

- *CAMPAGNE_NAME = My emailing*
- *CONTACT_LASTNAME = MARTIN*
- *CONTACT_FIRSTNAME = John*
- *CONTACT_SALUTATION = Mr*
- *Email = martin@naelan.com*

Pour le 2nd e-mail:

- *CAMPAGNE_NAME = My emailing*
- *CONTACT_LASTNAME = SIMONS*
- *CONTACT_FIRSTNAME = Marie*
- *CONTACT_SALUTATION = Mrs*
- *Email = simons@naelan.com*

Important notes:

- *Note that only the parent node of each element is to be defined*
- *It is possible to define several fields that point to the same element*
- *If one of the elements is iterative, the field takes the value of the element of the first iteration*
- *Each line ends with a line break (including the last line)*
- *In the case where the parent element exists several times in the flow, it is possible to specify the parent node of the parent node (example: SALUTATION_CONTACT; Salutation; OpportunityContactRole.ContactId;)*

- From the KSL Administration application, classify this mapping file in the KSL repository:
  - Select the Content repository tab
  - Add the mapping file as an external file and classify it in the tree structure

c. Step 3b: Create a style sheet for your new e-mailing template

If you already have a style sheet, go directly to the next step.

Otherwise, from KSL Studio (or KSL Office from KSL Server version 8.0.2.0), you can duplicate the FS_SALESFORCE_EMAIL style sheet and modify it to match your graphic charter.

d. Step 4b: Create the body of your e-mailing

From the Content repository tab of the KSL Administration application, create a group of text zones. This group will be the body of your e-mailing template.

This group of text zones contains one or more text zones that are also created from the Content Repository application. These text zones must use the style sheet created in the previous step 3.

The variables correspond to the Salesforce data are created in the text zone that require it.

- The value of each variable comes from the XML stream sent by Salesforce
- Each variable is in string format
- The name of each variable corresponds to the 1st element of the line of the mapping file that defines it

For the example described above, the following variables will be created:

- *CAMPAGNE_NAME*
- *CONTACT_LASTNAME*
- *CONTACT_FIRSTNAME*
- *CONTACT_SALUTATION*
- *Email*

e. Step 5b: Create a new KSL document template for the e-mailing template

In this context of creating a new e-mailing template, it is necessary to create a new KSL document template from which a new KSL Server service will be created in step 6.

The procedure from KSL Studio is as follows:

- Add the XSD file generated in step 1 to the KSL project classification plan
- Duplicate the generic edition SF_Emailing_500px and add it to the KSL project classification tree
- In this new document template, select the style sheet by selecting the style sheet component created in step 3 above
- Modify the XML request for this document template by specifying the new XSD
- Delete the group in the document template
- Insert the new text zone group created in step 4
- Save and Update and release the document template

f. Step 6b: Create the document service corresponding to the emailing template

- From KSL Admin, create the document service associated with the document template created in step 5
- Set a label in the description field: this description corresponds to the label displayed for this e-mail template in the list of templates presented to the Salesforce user in its Document Templates component.
- Classify this service in the project tree: the document service should be classified in a branch of the tree accessible to the Salesforce user in its Document Templates component accessible from the Campaign object.
- Specify the preview action group: EmailActions
- Fill in the parameters of this document service using the information below

Parameter	Default value or example	Description
feuillestyles (stylesheet)	FS_SALESFORCE_EMAIL	Select the style sheet used for editing (without extension) Deselect the option "Show this parameter in Ksl Office"
MailAccount	MARKETING	Specify the e-mail distribution account if KSL Server is the SMTP server, otherwise left this field blank This distribution account is used for mass e-mailing sending; it must match a distribution account created in KSL Admin Deselect the option "Show this parameter in Ksl Office"
MailBat	True	Set this parameter to True for this single on-demand e-mail sending case Deselect the option "Show this parameter in Ksl Office"
MailCharset	UTF-8	Set this parameter to UTF-8 *Encoding of the CSV file generated from the XML stream sent by the KSL plug-in* *Set this parameter by default to the universal UTF-8 code page; it is possible to change this encoding (example: UTF-8, Windows-1252, ISO8859-15 (Linux)* Deselect the option "Show this parameter in Ksl Office"
MailContentGroup	Example: <InvitationEvenement.ksg>	*Optional parameter for transmitting the group of text fields used as e-mail body* *If not specified, the group inserted by default in the document template is used* *This parameter requires that the document template dynamically calls the group (default group dynamically called with the expression MailContentGroup)*

		The group is defined between the characters < and > with the extension .ksg Deselect the option "Show this parameter in Ksl Office"
MailCsvFile	<default>	This file is used only if the P_MAP_FILE file is not defined. The default value can be used for debugging tests without XML stream. A value between the characters < and > indicates that the file is in the KSL repository. The CSV file is a text file with one contact / recipient per line, possibly with a header line that names the columns and column separators ";" You need to set a default value (for example "<default>") even if the P_MAP_FILE is defined Deselect the option "Show this parameter in Ksl Office"
MailCsvHeader		*Parameter not used in the case of an e-mailing* *This CSV is only used if P_MAP_FILE is not defined.* Deselect the option "Show this parameter in Ksl Office"
MailLinkFile	<emailing_links.csv>	Set if necessary the file giving the definition of hypertext links in the e-mail (see documentation on the KSL Email Designer function) Deselect the option "Show this parameter in Ksl Office"
MailReturnMode	2	Set this parameter to the value 2. Deselect the option "Show this parameter in Ksl Office" *For information:* - *1: for the creation of an e-mail sent by Salesforce* - *2: for the creation of an e-mailing sent by KSL Server or asynchronously sent by an external e-mail send & track provider*
MailTitle	Example: <CONTACT_SALUTATION> <CONTACT_LASTNAME> - Visit our trade show	Select a text only if you want the subject of the e-mails to be predefined You can enter a free text in and insert some variables defined in the mapping file Select or deselect the option "Show this parameter in Ksl Office", depending on whether you want the user to be able to change or not, the subject used when creating e-mailing
MailUrl	Example: https://www.naelan.fr/ksl_images/	Set the web directory Url used for storing images (see the KSL Email Designer documentation) Deselect the option "Show this parameter in Ksl Office"
P_FILE	Example: <SF_Emailing_Campaign.xml>	*XML data stream passed by the KSL for Salesforce plug-in* *This default value is used only during debugging tests*

		Deselect the option "Show this parameter in Ksl Office"
P_LABEL		*Parameter not used in the case of a single on demand e-mail sent by Salesforce®* Deselect the option "Show this parameter in Ksl Office"
language	Example: FR	Salesforce interface language passed through the KSL plug-in Deselect the option "Show this parameter in Ksl Office"
P_MAP_FILE	Example: <SF_emailing_mapping.map>	Indicate the mapping file created for the template and corresponding to its XSD Deselect the option "Show this parameter in Ksl Office"

g. Step 7b (optional): create other e-mailing templates based on the same data

In many business contexts, it is necessary to create a family of e-mailing templates that share the same graphic design and based on the same Salesforce objects and data.

In this case, it may be more efficient to create KSL e-mail services based on the same KSL e-mailing template. The procedure is then the following.

- In KSL Studio:
  - Edit the document template created in step 5
  - Reserve the template and edit the properties of the text zone group by right-clicking / Properties
  - In the window that appears, specify the expression MailContentGroup the Expression field of the dynamic group call
  - Validate, save and release the edition

- From the Content repository tab of the KSL Administration application,
  - Create as many groups of text boxes as desired body of e-mailing (1 group per model of e-mailing)

- In KSL Admin:
  - Recompile the document template above
  - Copy the document service as many times as needed (as much service as the family's email template)
  - Give a description to each copied service with the name of the template you want in Salesforce
  - Edit the functional parameters of each copied service and set the MailContentGroup parameter specifying the name of the group to call (Service 1> Group 1, Service 2> Group 2, etc.)

<u>Benefits:</u>

Thanks to this setting:

- All e-mailing templates share the same document template, the same document service, and the same style sheet

- Creating and editing an e-mailing body is done directly from Salesforce
- Adding a template is done in seconds by duplicating a service

2.7. Proposing an e-mail or e-mailing with alternative contents

KSL provides a powerful feature that allows to offer to users e-mail or e-mail templates with alternative content.

Thanks to this function, the user edits his e-mail or his e-mailing and can choose between several alternative text boxes in the KSL editor.

The SF_Emailing_500px service given as an example, implements this function and proposes in a single 4 e-mailings in a single template. The user selects the template "Family of e-mailings" and in a second time can select one of the 4 proposed contents corresponding to the groups:

- SF_InvitationEvenement.ksg
- SF_KSLEmailDesigner.ksg
- SF_NewsletterNaelan.ksg
- SF_Teasing_Solutions.ksg

Refer to the KSL Office User Guide for creating these alternative groups or text boxes.

NAELAN

Headquarter - 4 rue Claude Chappe
69370 Saint-Didier au Mont d'Or
France - Tél. +33 4 37 59 81 40

Paris Office - 4 Place Louis Armand
75023 Paris - France
Tél. +33 (0)1 72 76 80 86

www.naelan.com
contact@naelan.com
support@naelan.com