
Guidelines for Dynamics CRM Administrators

Dynamics CRM Connector Guidelines



officeatwork AG has prepared this manual with the greatest possible care so as to ensure that the information contained herein is easy to understand, accurate and reliable. Nevertheless, officeatwork AG is in no way liable for any issues which have any connection with this manual, including – and without restriction – its standard quality and availability for special purposes. From time to time, officeatwork AG will revise the software described in this manual and reserves the right to do so without prior advice to the customer. Under no circumstances is officeatwork AG liable for indirect, special or incidental damages resulting from the purchase or use of this manual or the information contained herein. This guarantee exclusion has no impact on the statutory rights of the user.

Copyright© 1992–2021 officeatwork AG, Switzerland.
All rights reserved.

officeatwork® is a registered trademark of officeatwork AG.

Microsoft® Word, Microsoft® Office, Windows®, Windows 95™, Windows 98™, Windows NT®, Windows XP®, Windows Vista, Windows 7, Windows 8, Windows 10 and MS-DOS™ are trademarks of the Microsoft Corporation.

Other names of companies, products or services may be trademarks or registered trademarks of the respective owners.

Table of Contents

About this guide	5
For whom is the guide intended.....	5
What is covered in this guide.....	5
Knowledge required.....	5
Typographic conventions.....	5
Overview	6
Benefit	6
Systems	6
Installation	7
Import Solutions	7
Security	7
Configuration.....	8
Menu integration	9
XML structure	9
XML Template	11
Procedure	11
Functions	13
Syntax.....	13
Functions	13
Processing.....	28
XML Web Resources	39
Use Cases.....	39
Appendix	41
OSCT examples	41
Start officeatwork Template Chooser	41
Create Letter to Contact.....	42
Create Letter to primary Contact of Account	43
Create project reports.....	45
Generate partner contract.....	47
Generate mass output for sales special	50
Support	53
Index	54

About this guide

For whom is the guide intended

This book has been written for Microsoft Dynamics CRM administrators that install officeatwork Dynamics CRM Connector on servers or online, for Dynamics CRM Developers or Power Users, which are writing XML Templates (OSCT and DCMLT files) and configuring the menu structure.

What is covered in this guide

This manual contains an installation and configuration guideline to setup the officeatwork Dynamics CRM Connector.

Knowledge required

You should be familiar with the general use and administration of Microsoft Dynamics CRM Server or Dynamics CRM Online.

Typographic conventions

Before reading this guide, you should be familiar with the typographic conventions used.

The following graphic descriptions highlight sections of text with particular significance.

Formatting Convention	Type of Information
Triangle ➤	Step-by-step procedure. You can follow these instructions to perform a specific task.
Bold Typeface	Objects needed for selection, such as menus, buttons, items in a list or table headers.
CAPITAL LETTERS	Key legends on the keyboard. For example SHIFT, CTRL or ALT.
KEY+KEY	Key combinations which must be pressed at the same time are marked with +. Examples: CTRL+P or ALT+F4.

CHAPTER 1

Overview

Benefit

With «officeatwork XML templates» (DCMLT- and OSCT-files), documents based on officeatwork templates can be created. Additionally Dynamics CRM data can be integrated directly into the documents with a defined syntax.

Systems

The following systems are provided for integration and use of officeatwork XML Templates:

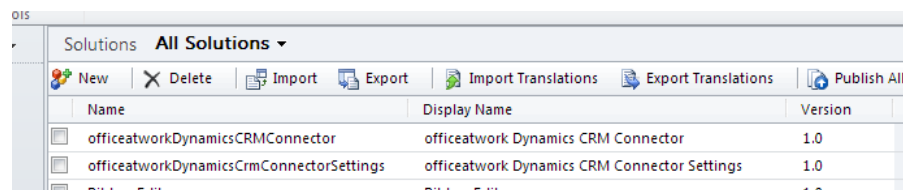
- Microsoft Dynamics CRM 2013, 2015 & 2016
- Enterprise Document Creation system:
 - Microsoft Office and officeatwork client
and/or
 - officeatwork EDC Server

Installation

Import Solutions

The officeatwork Dynamics CRM Connector is splitted into two Dynamics CRM solutions. The first solution “officeatwork Dynamics CRM Connector” includes all system/application components and the second solution “officeatwork Dynamics CRM Connector Settings” all components that can be customized to the customers requirements.

Both solutions have to be imported into your organization.



The screenshot shows the Dynamics CRM Solutions list. The table has columns for Name, Display Name, and Version. Two solutions are listed:

Name	Display Name	Version
officeatworkDynamicsCRMConnector	officeatwork Dynamics CRM Connector	1.0
officeatworkDynamicsCrmConnectorSettings	officeatwork Dynamics CRM Connector Settings	1.0

The solution “officeatwork Dynamics CRM Connector” needs to be imported as *managed* solution.

The solution “officeatwork Dynamics CRM Connector Settings” needs to be imported as *unmanaged* solution.

The solutions are optimized for Dynamics CRM Online installations. If you are working in an *on-premise* environment you have to change the isolation mode from sandbox to *none sandbox*.

Security

No special security is implemented for the officeatwork Dynamics CRM Connector. It is all based on the standard security of Microsoft Dynamics CRM.

Each user needs the permission to read and create new records of the entity "officeatwork Jobs". For a simple implementation the security role "officeatwork User" is installed. Each officeatwork user should be member of this security role. Therefore, you only have to grant the permission to create and read elements of the entity officeatwork Jobs to this security role.

Security Role: officeatwork User									
Entity	Create	Read	Write	Delete	Append	Append To	Assign	Share	
Backlog Item	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Case Contact	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Communication	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Connect	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CV	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Influencer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Invoice Position	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
InvoiceX	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Issue Tracker	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Licence	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
officeatwork Jobs	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Partner Contract	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Configuration

The configuration is splitted into two parts - the dynamic menu structure (detailed explanation in chapter “Menu integration”) and the use of officeatwork XML Templates (detailed explanation in chapter “XML Templates”). Both can be configured in the webresources.

CHAPTER 3

Menu integration

XML structure

One web resource called “offatwrk_dynamicmenupopulator.xml” is imported with the «officeatwork Dynamics CRM Connector Settings». It’s a subpart of the ribbon.xml definition of the Dynamics CRM Ribbon.

Description

This xml describes the menu structure for each entity. You have to define the web resource name of the web resource with the a officeatwork XML Template. Additionally, a label text and alternative text for the button have to be defined here.

For each entity a “Menu Id” section has to be added. Inside of the “Controls Id” section multiple “Button Id” elements can be added.

The buttons are ordered according to the Sequence number.

Syntax

```
<?xml version="1.0"?>
<Menus>
  <Menu Id="Entityname">
    <MenuSection Id="officeatwork.Dynamic.MenuSection" Sequence="10">
      <Controls Id="officeatwork.Dynamic.Controls">
        <Button Id="[Webresource name]" Command="officeatwork.CreateJob"
          Sequence="[Sequence number]" LabelText="[Label text]" Alt="[Alt text]" />
        ...
      </Controls>
    </MenuSection>
  </Menu>
  ...
</Menus>
```

Parameter

Entityname	The name of the entity in which the button should be displayed
Webresource name	The name of the linked xml webresource with the officeatwork XML Template as content
Sequence number	The sequence of the buttons in the menu
Label text	The text shown on the button
Alt text	The text shown in the alternative text of the button

Example

In this example two dynamic Buttons on the entity «contact» ribbon and three buttons on the entity «account» ribbon are shown. The two linked web resources from the contact buttons are equivalent to two of the three account buttons (Letter and Fax). The account ribbon has an additional third button used for protocol creation.

```
<?xml version="1.0"?>
<Menus>
  <Menu Id="Contact">
    <MenuSection Id="officeatwork.Dynamic.MenuSection" Sequence="10">
      <Controls Id="officeatwork.Dynamic.Controls">
        <Button Id="Letter.osct" Command="officeatwork.CreateJob" Sequence="20" LabelText="Letter" Alt="Create
letter for contact" />
        <Button Id="Fax.osct" Command="officeatwork.CreateJob" Sequence="30" LabelText="Fax" Alt="Create a fax for
contact" />
      </Controls>
    </MenuSection>
  </Menu>
  <Menu Id="[Entityname]">
    <MenuSection Id=" officeatwork.Dynamic.MenuSection" Sequence="10">
      <Controls Id=" officeatwork.Dynamic.Controls">
        <Button Id="Letter.osct" Command="officeatwork.CreateJob" Sequence="20" LabelText="Letter for account"
Alt="Create a letter for account" />
        <Button Id="Fax.osct" Command="officeatwork.CreateJob" Sequence="30" LabelText="Fax for account"
Alt="Create a fax for account" />
        <Button Id="Protocol.dcm1t" Command="officeatwork.CreateJob" Sequence="40" LabelText="Protocoll for
account" Alt="Create a protocol" />
      </Controls>
    </MenuSection>
  </Menu>
</Menu>
</Menu>
</Menus>
```

CHAPTER 4

XML Template

Procedure

For an officeatwork XML Template (DCMLT- or OSCT-File) creation the xml text has to be written with a text editor.

Here you can find an example for a simple implementation to add an OSC-File in the notes of a contact to create a letter:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Parameters>

  <CreateDocument>
    <TemplateID>letter</TemplateID>
  </CreateDocument>

  <Processing>

    <Logging>
      <OutputFiles>
        <OscInclude>True</OscInclude>
        <OscInclude>False</OscInclude>
      </OutputFiles>
    </Logging>

    <DestinationSets>
      <DestinationSet>
        <Filters>
          <Filter Type="FileName">*.osc</Filter>
        </Filters>
        <Destinations>
          <Destination Type="DynamicsCrmNote">
            <Relationships>
              <Relationship>
                <Entity>contact</Entity>
                <Guid>{{Field('contactid')}}</Guid>
              </Relationship>
            </Relationships>
          </Destination>
        </Destinations>
      </DestinationSet>
    </DestinationSets>

  </Processing>
</Parameters>
```

This example creates a new document based on the Master-Template «Letter» enclosed in your officeatwork solution without any Dynamics CRM data integration.

Detailed syntax descriptions can be found in the next chapter.

If no encoding is defined in the XML header, the default value for the encoding is UTF-8.

The XML Template Type has to be the last part of the name of the webresource (*.dcmlt or *.osct)

After writing the xml code for the XML Template, it can be copied into a web resource of type data (xml) in your organization.

This is an example for a Letter OSCT file in the web resources:

Web Resource: New - Microsoft Dynamics CRM - Google Chrome

https://officeatwork3.crm4.dynamics.com/main.aspx?appSolutionId=%7bFD140AAF-4DF4-11DD

Microsoft Dynamics CRM Roman Stalder officeatwork

File Web Resource

Save Save and Close

Save Record Management Actions

Solution: Default Solution

Web Resource: New

General

Name * new_Letter.osct

Display Name Letter OSCT

Description

Content

Type * Data (XML) Text Editor

Language

Upload File Choose File No file chosen

URL

URL

Save and publish the webresource afterwards.

Functions

In this chapter the available functions for data integration from Microsoft Dynamics CRM are explained including its syntax.

Syntax

For building XML Template the CRM Connectro offers various functions with the following structure:

- All functions are enclosed by two leading {{ and two closing }} brackets
- The parameters are defined inside of () parentheses
- Every parameter has a leading and a closing apostrophe "
- The parameters are separated by a semicolon ;

This is prerequisite for the proper execution of its functions:

Example

```
{{Function('Parameter1'; 'Parameter2')}}
```

All of the following functions return a value in string format if not otherwise specified.

Functions

Field()

Description

With the Field() function you can get the data from one field of the calling entity.

Syntax

```
{{Field('FieldName'; ['DefaultValue'])}}
```

Parameter

FieldName	Name of the field that contains the needed information
DefaultValue	Optional default return value if the field does not exist Default: "" (String.Empty)

Example 1

The calling entity for this example is the account entity.

```
Function:    {{Field('name')}}
```

```
Result:     officeatwork AG
```

Example 2

The calling entity for this example is an entity without the defined field.

Function: `{{Field('name', 'no name defiend')}}}`

Result: no name defiend

FetchXml ()

Description

The FetchXml() function gets the xml serialized result of the executed FetchXml statement.

Syntax

```
{{FetchXml ('FetchXmlQuery')}}}
```

Parameter

FetchXmlQuery A valid FetchXml query expression

All xml namespaces in the serialized response will be removed.

This function is only available when the connector in the non-sandboxed mode is installed.

For more information of constructing FetchXml queries read the official documentation from Microsoft.

Build queries with FetchXml: <http://msdn.microsoft.com/en-us/library/gg328332.aspx>

Example

In this example we fetch all accounts owned by the user Peter Muster.

```
Function:  {{FetchXml('<fetch version="1.0" output-format="xml-platform"
mapping="logical" distinct="false">
  <entity name="account">
    <attribute name="name" />
    <attribute name="accountnumber" />
    <filter type="and">
      <condition attribute="statecode" operator="eq" value="0" />
      <condition attribute="ownerid" operator="eq" uiname="Peter
Muster" uitype="systemuser" value="{CAC33D49-D3D4-E311-
A9EB-D89D6763DDB0}" />
    </filter>
  </entity>
</fetch>')}}}
```

```
Result:  <ArrayOfEntity>
  <Entity>
    <Attributes>
      <KeyValuePairOfstringanyType>
        <key>name</key>
        <value type="d5pl:string">Name of Account1</value>
      </KeyValuePairOfstringanyType>
      <KeyValuePairOfstringanyType>
        <key>accountnumber</key>
        <value type="d5pl:string">Account1.com</value>
      </KeyValuePairOfstringanyType>
    </Attributes>
    <EntityState nil="true" />
    <FormattedValues />
    <Id>d8d78a0b-6ca6-14ee-a747-d89d6742fc7b</Id>
    <LogicalName>account</LogicalName>
    <RelatedEntities />
  </Entity>
  <Entity>
    <Attributes>
      <KeyValuePairOfstringanyType>
        <key>name</key>
        <value type="d5pl:string">Name of Account2</value>
      </KeyValuePairOfstringanyType>
```

```

        <KeyValuePairOfstringanyType>
          <key>accountnumber</key>
          <value type="d5p1:string">Account2.ch</value>
        </KeyValuePairOfstringanyType>
      </Attributes>
      <EntityState nil="true" />
      <FormattedValues />
      <Id>d8d78a0b-6cb2-c412-a733-d77d6763dd78</Id>
      <LogicalName>account</LogicalName>
      <RelatedEntities />
    </Entity>
  </ArrayOfEntity>

```

FetchXmlCount()

Description

With the FetchXmlCount() function the amount of the resulting records of the FetchXml query can be counted.

Syntax

```
{{FetchXmlCount('FetchXmlQuery')}}}
```

Parameter

FetchXmlQuery A valid FetchXml query

Example

In this example we count the amount of contacts with country switzerland.

```

Function:   {{FetchXmlCount('<fetch mapping="logical" aggregate="false"
distinct="true">
  <entity name="contacts">
    <attribute name="lastname" />
    <filter>
      <condition attribute="address1_country" operator="eq"
value="Switzerland" />
    </filter>
  </entity>
</fetch>')}}}
Result:    712

```

FetchXmlLoop()

Description

With the FetchXmlLoop() function all records of the FetchXml query can be looped.

Syntax

```
{{FetchXmlLoop('LoopId'; 'FetchXmlQuery'; 'Content')}}}
```

Parameter

LoopId Unique ID for loop
FetchXmlQuery A valid FetchXml query
Content Content that's processed through the loop

Example

In this example we get the first and the last name of all contacts with country equal Switzerland ordered by the last name ascending.

```
Function:    {{FetchXmlLoop('ContactsLoop'; '<fetch mapping="logical"
              aggregate="false" distinct="true">
              <entity name="contact">
                <attribute name="lastname" />
                <attribute name="firstname" />
                <order attribute="lastname" descending="false" />
                <filter>
                  <condition attribute="adress1_country" operator="eq"
                    value="Switzerland" />
                </filter>
              </entity>
            </fetch>; '{{FetchXmlLoopField('ContactsLoop'; 'lastname')}}
            {{FetchXmlLoopField('ContactsLoop'; 'firstname')}}}}}
```

```
Result:    Schneider Marcel
           Stadelmann Ramon
           Winiger Rene
           ...
```

FetchXmlLoopField()

Description

With the FetchXmlLoopField() function you can get the value of a field during a loop over all records of the FetchXml query.

Syntax

```
{{FetchXmlLoopField('LoopId'; 'FieldName';
['GetReference']; ['DefaultValue']}}
```

Parameter

LoopId	Unique ID to related loop
FieldName	Field name of the field in the related entity
GetReference	Optional boolean value to get the reference (not the value) of a field (works only with lookup fields) Default: false
DefaultValue	Optional default value if the field does not exist in the result fo the FetchXml query Default: "" (string.Empty)

The function FetchXmlLoopField() is only available inside a FetchXmlLoop().

Example 1

In this example we get the first and last name of all contacts with country equal Switzerland sorted by lastname ascending.

```
Function:    {{FetchXmlLoop('ContactsLoop'; '<fetch mapping="logical"
              aggregate="false" distinct="true">
              <entity name="contact">
                <attribute name="lastname" />
                <attribute name="firstname" />
                <order attribute="lastname" descending="false" />
                <filter>
                  <condition attribute="adress1_country" operator="eq"
                    value="Switzerland" />
                </filter>
```



```

    </entity>
  </fetch>'; '{{FetchXmlLoopField('ContactsLoop'; 'lastname')}}
  {{FetchXmlLoopField('ContactsLoop'; 'firstname')}})'))}}

```

Result: Schneider Marcel
Stadelmann Ramon
Winiger Rene
...

Example 2

In this example we get the first and last name of all contacts with country equal Switzerland sorted by lastname ascending. Additionally, we get the reference (parameter GetReference = true) for the account.

```

Function:  {{FetchXmlLoop('ContactsLoop'; '<fetch mapping="logical"
          aggregate="false" distinct="true">
            <entity name="contact">
              <attribute name="lastname" />
              <attribute name="firstname" />
              <attribute name="new_language" />
              <order attribute="lastname" descending="false" />
              <filter>
                <condition attribute="adress1_country" operator="eq"
                  value="Switzerland" />
              </filter>
            </entity>
          </fetch>'; '{{FetchXmlLoopField('ContactsLoop'; 'lastname')}}
          {{FetchXmlLoopField('ContactsLoop'; 'firstname')}}},
          {{FetchXmlLoopField('ContactsLoop'; 'new_language'; 'true')}})'))}}

```

Result: Schneider Marcel {68F15588-1DAB-E111-875A-18A905732A15}
Stadelmann Ramon {50ED1CCE-0B25-E111-93DC-1CC1DE6D1BD7}
Winiger Rene {779B3CE7-0B25-E111-93DC-1CC1DE6D1BD7}
...

Example 3

In this example we get the first and last name of all contacts with country equal Switzerland sorted by lastname ascending. Additionally, we get the reference (parameter GetReference = true) for the account. If the contact has no defined account, we want to get the value "no account defined"

```

Function:  {{FetchXmlLoop('ContactsLoop'; '<fetch mapping="logical"
          aggregate="false" distinct="true">
            <entity name="contact">
              <attribute name="lastname" />
              <attribute name="firstname" />
              <attribute name="new_language" />
              <order attribute="lastname" descending="false" />
              <filter>
                <condition attribute="adress1_country" operator="eq"
                  value="Switzerland" />
              </filter>
            </entity>
          </fetch>'; '{{FetchXmlLoopField('ContactsLoop'; 'lastname')}}
          {{FetchXmlLoopField('ContactsLoop'; 'firstname')}}},
          {{FetchXmlLoopField('ContactsLoop'; 'account'; 'true'; 'no account
          defined')}})'))}}

```

Result: Schneider Marcel {68F15588-1DAB-E111-875A-18A905732A15}
Stadelmann Ramon no account defined
Winiger Rene {779B3CE7-0B25-E111-93DC-1CC1DE6D1BD7}
...

FetchXmlLoopSum()

Description

With the FetchXmlLoopSum() function you can get the sum of a field over all entities from the FetchXml query.

Syntax

```
{{FetchXmlLoopSum('LoopId'; 'FetchXmlQuery'; ['FieldNames'])}}
```

Parameter

LoopId	Unique ID to related loop
FetchXmlQuery	A valid FetchXml query
FieldNames	Comma separates name of the fields to calculate the sum.

The function FetchXmlLoopSum() is only available inside a FetchXmlLoop().

If you aggregate a CRM currency field with the standard fetch xml functionality, the crm calculates the result in the defined base currency of the organization. In some cases you need the value in the actual currency and not in the base currency. For this case you can use this function.

Be careful, because all currency values are in their actual currency!

Example

In this example we loop over all products filtered by the currency. We get the product type name (like cars, motorcycles or bicycles) and the total sum of the prices over all products of the same product type. The output is ordered ascending by the product type name.

```
Function:    {{FetchXmlLoop('ProductsLoop'; '<fetch mapping="logical"
aggregate="false" distinct="true">
  <entity name="products">
    <attribute name="product_type" />
    <attribute name="product_typedname" />
    <order attribute="product_typedname" descending="false" />
    <filter>
      <condition attribute="transactioncurrencyid"
operator="eq" value="5737BD56-E0D7-E111-8D12-
984BE17C38F7" />
    </filter>
  </entity>
</fetch>'; '{{FetchXmlLoopField('ProductsLoop';
'product_typedname')}} {{FetchXmlLoopSum('ProductsLoop'; '<fetch
mapping="logical" aggregate="false" distinct="false">
  <entity name="products">
    <attribute name="product_price" />
    <filter>
      <condition attribute="product_type"
operator="eq" value="{{FetchXmlLoopField('ProductsLoop';
'product_typedname')}}" />
    </filter>
  </entity>
</fetch>')}}'}}
Result:    Bicycles 12084.60
           Cars 411975.00
           Motorcycles 207813.15
           ...
```

FetchXmlLoopAvg()

Description

With the FetchXmlLoopAvg() function the average of a field over all records from the FetchXmlQuery result can be gotten.

Syntax

```
{{FetchXmlLoopAvg('LoopId'; 'FetchXmlQuery'; ['FieldNames'])}}
```

Parameter

LoopId	Unique ID to related loop
FetchXmlQuery	A valid FetchXml query
FieldNames	Comma separates the names of the fields to calculate the average

The function `FetchXmlLoopAvg()` is only available inside a `FetchXmlLoop()`.

If you aggregate a CRM currency field with the standard fetch xml functionality, the crm calculates the result in the defined base currency of the organization. In some cases you need the value in the actual currency and not in the base currency. In this case this function can be used.

Be careful, because all currency values are in their actual currency!

Example

In this example we loop over all products filtered by the currency. We get the product type name (like cars, motorcycles or bicycles) and the average price of all products of the same product type. The output is ordered ascending by the product type name.

```
Function:    {{FetchXmlLoop('ProductsLoop'; '<fetch mapping="logical"
aggregate="false" distinct="true">
  <entity name="products">
    <attribute name="product_type" />
    <attribute name="product_typedisplayname" />
    <order attribute="product_typedisplayname " descending="false" />
    <filter>
      <condition attribute="transactioncurrencyid"
operator="eq" value="5737BD56-E0D7-E111-8D12-
984BE17C38F7" />
    </filter>
  </entity>
</fetch>'; '{{FetchXmlLoopField('ProductsLoop';
'product_typedisplayname')}} {{FetchXmlLoopAvg('ProductsLoop'; '<fetch
mapping="logical" aggregate="false" distinct="true">
  <entity name="products">
    <attribute name="product_price" />
    <filter>
      <condition attribute="product_type"
operator="eq" value="{{FetchXmlLoopField('ProductsLoop';
'product_type')}}" />
    </filter>
  </entity>
</fetch>')}}')}}
Result:    Bicycles 805.64
           Cars 27465.00
           Motorcycles 13854.21
           ...
```

JobField()**Description**

With the `JobField()` function you can get the data from one field of the created `offatwrk_job` entity.

Syntax

```
{{JobField('FieldName'; ['DefaultValue'])}}
```

Parameter

FieldName	Name of the field that contains the needed information
DefaultValue	Optional default return value if the field does not exist Default: "" (String.Empty)

Example 1

Getting the GUID of the created job entity.

Function: `{{JobField('offatwrk_jobid')}}}`

Result: `65c193fd-e63f-e511-811c-c4346bac0bdc`

Example 2

Getting the template name of the created job entity.

Function: `{{JobField('offatwrk_templatename', 'no template defiend')}}}`

Result: `new_LetterToContact.osct`

JobLoop()

Description

With the JobLoop() function you can loop over all records in the field “Calling entity GUID’s” in the officeatwork_jobs entity.

Syntax

```
{{JobLoop('LoopId'; ['OrderFieldName']; ['OrderAscending'];  
'Content']}}}
```

Parameter

LoopId	Unique ID for loop
OrderFieldName	Optional field name of the field in the related entity to sort/order with Default: "" (no order/sort)
OrderAscending	Optional sort/order direction to the OrderFieldName This parameter is also needed when the OrderFieldName parameter contains a value Default: "true" (order ascending)
Content	Content to proceed during the loop

The function JobLoop() is only available if the flag «single mode» is set in the officeatwork_jobs entry.

Example 1

The calling entity for this example is the account entity. And there is a loop over all ticked accounts in the account overview.

Function: `{{JobLoop('JobAccountLoop'; '{{JobLoopField('JobAccountLoop'; 'accountid')}} - {{JobLoopField('JobAccountLoop'; 'name')}}')}}}`

Result: `muster.com - Muster SA
example.org - Example Corporation
officeatwork.com - officeatwork Ltd.
...`

Example 2

The calling entity for this example is the account entity. And there is a loop over all ticked accounts in the account overview. The accounts are sorted by the accountid ascending.

Function: `{{JobLoop('JobAccountLoop'; 'accountid'; 'true';
'{{JobLoopField('JobAccountLoop'; 'accountid')}} -
'{{JobLoopField('JobAccountLoop'; 'name')}}')}}}`

Result: `example.org - Example Corporation
muster.com - Muster SA`

```
officeatwork.com - officeatwork Ltd.
...
```

JobLoopField()

Description

With the JobLoopField() function you can get the value of a field in an entity during a loop over all records in the field “Calling entity GUID’s” in the officeatwork_jobs entity.

Syntax

```
{{JobLoopField('LoopId'; 'FieldName'; ['DefaultValue'])}}
```

Parameter

LoopId	Unique ID to related loop
FieldName	Name of the field in the related entity
DefaultValue	Optional default value if the field does not exist in the related entity Default: "" (string.Empty)

The function JobLoopField() is only available inside a JobLoop().

Example 1

The calling entity for this example is the account entity. And there is a loop over all ticked accounts in the account overview.

```
Function:  {{JobLoop('JobAccountLoop'; '{{JobLoopField('JobAccountLoop';  
          'accountid')}} - {{JobLoopField('JobAccountLoop'; 'name')}}')}}}
```

```
Result:   muster.com - Muster SA  
          example.org - Example Corporation  
          officeatwork.com - officeatwork Ltd.  
          ...
```

Example 2

The calling entity for this example is the account entity. And there is a loop over all ticked accounts in the account overview. The accounts are sorted by the accountid ascending.

```
Function:  {{JobLoop('JobAccountLoop'; 'accountid'; 'true';  
          '{{JobLoopField('JobAccountLoop'; 'accountid')}} -  
          {{JobLoopField('JobAccountLoop'; 'name')}}')}}}
```

```
Result:   example.org - Example Corporation  
          muster.com - Muster SA  
          officeatwork.com - officeatwork Ltd.  
          ...
```

If()

Description

Depending on the truth value of the condition, the function returns the value of the parameter [Value when true] or the [Value when false].

Syntax

```
{{If('Condition'; 'Value when true'; ['Value when false'])}}
```

Parameter

Condition	Condition which is necessary for examination. It's composed of an operator and two numbers. Available operators: [<code><=</code> <code>>=</code> <code>==</code> <code>!=</code> <code><</code> <code>></code>]
Value when true	Return value of the function when the condition is true.
Value when false	Optional return value of the function when the condition is false. Default: "" (String.Empty)

Example 1

```
Function:    {{If('7!=6'; 'Right! 7 is not equal 6!')}}  
Result:     'Right! 7 is not equal 6!'
```

Example 2

```
Function:    {{If('7!=6'; 'Right! 7 is not equal 6!'; 'False, 7 is equal 6!')}}  
Result:     'Right! 7 is not equal 6!'
```

Example 3

```
Function:    {{If('7==6'; 'Right! 7 is equal 6?!'; 'False, 7 is not equal 6!')}}  
Result:     False, 7 is not equal 6!
```

Example 4

```
Function:    {{If('{{RelatedCount('contacts'; 'accountid')}}>0'; 'This account  
has related contacts'; 'There are no contacts with a relation to  
this account')}}  
Result:     This account has related contacts
```

Today()

Description

With the Today() function you can get the actual start date of the document creation.

Syntax

```
{{Today(['Format'])}}
```

Parameter

Format	Optional format of the result Default: "DD.MM.YYYY"
--------	--

For more information of formats read the official documentation from Microsoft.

Custom Date and Time Format Strings: [https://msdn.microsoft.com/en-us/library/8kb3ddd4\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/8kb3ddd4(v=vs.100).aspx)

Example 1

The calling entity for this example is the account entity.

```
Function:    {{Today()}}
```

```
Result:     21.05.2015
```

Example 2

The calling entity for this example is the account entity.

```
Function:    {{Today('MM-dd-yy')}}
```

```
Result:     05-21-15
```

Now()**Description**

With the Now() function you can get the actual start date time of the document creation.

Syntax

```
{{Now(['Format'])}}
```

Parameter

Format	Optional format of the result Default: "DD.MM.YYYY"
--------	--

Example 1

The calling entity for this example is the account entity.

```
Function:    {{Now()}}
```

```
Result:     21.05.2015
```

Example 2

The calling entity for this example is the account entity.

```
Function:    {{Now('dd.MM.yyyy hh:mm:ss')}}
```

```
Result:     21.05.2015 11:23:47
```

RelatedField()**Description**

With the RelatedField() function you can get the value of a field of a related record of the entity.

Syntax

```
{{RelatedField('KeyFieldName'; 'EntityName'; 'FieldName';  
['DefaultValue'])}}
```

Parameter

KeyFieldName	Name of the field in the calling entity with the foreign key to the related entity
EntityName	Name of the related entity of the calling entity
FieldName	Name of the field in the related entity
DefaultValue	Optional default value if the field does not exist in the related entity Default: "" (string.Empty)

Example 1

The calling entity for this example is the account entity.

```
Function:  {{RelatedField('new_ceocontact'; 'contact'; 'lastname')}}  
Result:   Schneider
```

Example 2

The calling entity for this example is the account entity.

```
Function:  {{RelatedField('new_ceocontact'; 'contact'; 'lastname'; 'no name  
defined')}}  
Result:   Schneider
```

RelatedCount()

Description

With the RelatedCount() function you can count the amount of related records of the defined entity.

Syntax

```
{{RelatedCount('EntityName'; 'KeyFieldName'; ['Distinct'])}}
```

Parameter

EntityName	Name of the related entity to the calling entity
KeyFieldName	Name of the field in the related entity with the foreign key to the calling entity
Distinct	Optional boolean value that operates on the primary key of the entity instances. Use this field with linked entities to avoid multiple identical records in the query results. Use true to return no duplicate instances of the entities; otherwise, use false. Default: "false" (no distinct)

Example 1

The calling entity in this example is the account entity.

```
Function:  {{RelatedCount('contact'; 'accountid')}}  
Result:   23
```

Example 2

The calling entity in this example is the account entity. Additionally the contacts will be distinct by the primary key.

```
Function:  {{RelatedCount('contact'; 'accountid'; 'true')}}  
Result:   17
```


RelatedLoop()

Description

With the RelatedLoop() function you can loop over all related records of an entity.

Syntax

```
{{RelatedLoop('LoopId'; 'EntityName'; 'KeyFieldName';
['OrderFieldName']; ['OrderAscending']; ['Distinct']); 'Content'}}
```

Parameter

LoopId	Unique ID for loop
EntityName	Name of the related entity of the calling entity
KeyFieldName	Name of the field in the related entity with the foreign key to the calling entity
OrderFieldName	Optional field name of the field in the related entity to sort/order with Default: "" (no order/sort)
OrderAscending	Optional sort/order direction to the OrderFieldName This parameter is also needed when the OrderFieldName parameter contains a value Default: "true" (order ascending)
Distinct	Optional boolean value that operates on the primary key of the entity instances. Use this field with linked entities to avoid multiple identical records in the query results. Use true to return no duplicate instances of the entities; otherwise, use false. Default: "false" (no distinct)
Content	Content to proceed during the loop

Example 1

The calling entity for this example is the account entity. And there is a loop over all related contacts.

```
Function:    {{RelatedLoop('ContactLoop'; 'contact'; 'accountid';
              '{{RelatedLoopField('ContactLoop'; 'lastname')}}
              {{RelatedLoopField('ContactLoop'; 'firstname')}})}}
Result:     Schneider Marcel
            Winiger Rene
            Stadelmann Ramon
            ...
```

Example 2

The calling entity for this example is the account entity. And there is a loop over all related contacts. The contacts are sorted by lastname descending.

```
Function:    {{RelatedLoop('ContactLoop'; 'contact'; 'accountid'; 'lastname';
              'false'; '{{RelatedLoopField('ContactLoop'; 'lastname')}}
              {{RelatedLoopField('ContactLoop'; 'firstname')}})}}
Result:     Winiger Rene
            Stadelmann Ramon
            Schneider Marcel
            ...
```

Example 3

The calling entity for this example is the account entity. And there is a loop over all related contacts. The contacts are sorted by department descending and distinct.

```

Function:    {{RelatedLoop('ContactLoop'; 'contact'; 'accountid';
              'new_departement'; 'false'; 'true';
              '{{RelatedLoopField('ContactLoop'; 'new_departement')}}')}}
Result:     Development
            Education
            Marketing
            ...

```

RelatedLoopField()

Description

With the RelatedLoopField() function you can get the value of a field in a related entity during a loop over all related records of the calling entity.

Syntax

```
{{RelatedLoopField('LoopId'; 'FieldName'; ['DefaultValue'])}}
```

Parameter

LoopId	Unique ID to related loop
FieldName	Name of the field in the related entity
DefaultValue	Optional default value if the field does not exists in the related entity Default: "" (string.Empty)

The function RelatedLoopField() is only available inside a RelatedLoop().

Example 1

The calling entity for this example is the account entity. And there is a loop over all related contacts.

```

Function:    {{RelatedLoop('ContactLoop'; 'contact'; 'accountid';
              '{{RelatedLoopField('ContactLoop'; 'lastname')}}
              '{{RelatedLoopField('ContactLoop'; 'firstname')}}')}}
Result:     Schneider Marcel
            Winiger Rene
            Stadelmann Ramon
            ...

```

Example 2

The calling entity for this example is the account entity. And there is a loop over all related contacts. The contacts are sorted by lastname descending.

```

Function:    {{RelatedLoop('ContactLoop'; 'contact'; 'accountid'; 'lastname';
              'false'; '{{RelatedLoopField('ContactLoop'; 'lastname')}}
              '{{RelatedLoopField('ContactLoop'; 'firstname')}}')}}
Result:     Winiger Rene
            Stadelmann Ramon
            Schneider Marcel
            ...

```

Example 3

The calling entity for this example is the account entity. And there is a loop over all related contacts. The contacts are sorted by department descending and distinct.

```

Function:    {{RelatedLoop('ContactLoop'; 'contact'; 'accountid';
              'new_departement'; 'false'; 'true';
              '{{RelatedLoopField('ContactLoop'; 'new_departement')}}')}}

```

```
Result:      Development
            Education
            Marketing
            ...
```

XmlWebResourceValue()

Description

With the XmlWebResourceValue() function you can read values from web resources (xml format).

Syntax

```
{{XmlWebResourceValue('WebresourceName'; 'XPath';
['DefaultValue'])}}
```

Parameter

WebresourceName	Name of the web resource
xPath	xPath instruction to execute
DefaultValue	Optional default return value if the XPath does not exist Default: "" (String.Empty)

Example

In the following example we read the currency value translated to swiss german for the swiss currency from the web resource "new_translations.xml" (written in xml format).

```
Function:    {{XmlWebResourceValue('new_translations.xml';
    '//Translations/Label[@Name="Currency_CHF"]/Translation[@LCID=2055]
    ')}}
```

```
Result:      Franken
```

Processing

The XML parameter consists of easy-to-follow instructions that are processed when they are passed to the officeatwork connector. The following simple XML file is a sample that creates a letter in a Word-Document (DOCX) on an EDC-Server instance and adds the document as an annotation to an existing contact entity in Microsoft Dynamics CRM.

In this document the whole content of the <Processing> tag is explained. For detailed information to the <CreateDocument> tag and its content please read the officeatwork API document for developers.

Examples

DCMLT example

In an DCMLT file the <Processing> element is the child element of the <Data> element.

```
<?xml version="1.0" encoding="UTF-8"?>
<DCML>

  <Instruction>
    <CreateDocument>
      <TemplateID>Letter</TemplateID>
      <Language>2055</Language>
      <Output>
        <Save Filename="Letter.docx" Fileformat="DOCX" DeliveryMethod="Download" />
      </Output>
    </CreateDocument>
  </Instruction>

  <Data>
    <Processing>

      <Logging>
        <OutputFiles>
          <XmlInclude>True</XmlInclude>
          <XsltInclude>True</XsltInclude>
        </OutputFiles>
      </Logging>

      <Rendering Type ="EdcServer">
        <Url>http://ourserver/officeatwork/edcserver</Url>
      </Rendering>

      <DestinationSets>
        <DestinationSet>
          <Filters>
            <Filter Type="FileName">*. *</Filter>
          </Filters>
          <Destinations>
            <Destination Type="DynamicsCrmNote">
              <Relationships>
                <Relationship>
                  <Entity>contact</Entity>
                  <Guid>{{Field('contactid')}}</Guid>
                </Relationship>
              </Relationships>
            </Destination>
          </Destinations>
        </DestinationSet>
      </DestinationSets>

    </Processing>
  </Data>

</DCML>
```

OSCT example

In an OSCT file the <Processing> element is the child element of the < Parameters > element.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<Parameters>

  <CreateDocument>
    <TemplateID>Letter</TemplateID>
    <Language>2055</Language>
    <Output>
      <Save Filename="Letter.docx" Fileformat="DOCX" DeliveryMethod="Download" />
    </Output>
  </CreateDocument>

  <Processing>

    <Logging>
      <OutputFiles>
        <OscInclude>True</OscInclude>
        <OsctInclude>False</OsctInclude>
      </OutputFiles>
    </Logging>

    <Rendering Type ="EdcServer">
      <Url>http://ourserver/officeatwork/edcserver</Url>
    </Rendering>

    <DestinationSets>
      <DestinationSet>
        <Filters>
          <Filter Type="FileName">*. *</Filter>
        </Filters>
        <Destinations>
          <Destination Type="DynamicsCrmNote">
            <Relationships>
              <Relationship>
                <Entity>contact</Entity>
                <Guid>{{Field('contactid')}}</Guid>
              </Relationship>
            </Relationships>
          </Destination>
        </Destinations>
      </DestinationSet>
    </DestinationSets>

  </Processing>
</Parameters>

```

<Logging>

Description

The <Logging> element is a sub element of the <Processing> element and can contain sub elements.

Syntax

```

<Logging>
</Logging>

```

Example

The following example adds the XML Template from the web resource and the generated XML file to the collection of files for output (output files).

```

Function:      <Processing>
                <Logging>
                  <OutputFiles>
                    <XmlInclude>True</XmlInclude>
                    <XsltInclude>True</XsltInclude>
                  </OutputFiles>
                </Logging>
                ...
            </Processing>

```

<OutputFiles>

Description

The <OutputFiles> is a sub element of the <Logging> element and can contain sub elements.

Syntax

```
<OutputFiles>  
</OutputFiles>
```

Example

The following example adds the generated xml file to the collection of files that can be used in the <DestinationSets>.

```
Function:    <Logging>  
            <OutputFiles>  
              <XmlInclude>True</XmlInclude>  
              <XmltInclude>False</XmltInclude>  
            </OutputFiles>  
          </Logging>
```

<XmlInclude>

Description

The <XmlInclude> is a sub element of the <OutputFiles> with a boolean value. This element can't contain sub elements.

Syntax

```
<XmlInclude>[Value]</XmlInclude>
```

Parameter

Value	True: the generated xml file will be added to the outputfiles collection False: the generated xml file won't be added to the outputfiles collection Default: False
-------	--

Example 1

The following example adds the generated xml file to the collection of files that can be used in the <DestinationSets> (outputfiles). If that is done, the file can be saved in the destinationsets to a destination.

```
Function:    <OutputFiles>  
            <XmlInclude>True</XmlInclude>  
          </OutputFiles>
```

Example 2

The following example doesn't add the generated xml file to the collection of outputfiles. If that is done, the file cannot be in the destinationsets to a destination.

```
Function:    <OutputFiles>
             <XmltInclude>False</XmltInclude>
             </OutputFiles>
```

<XmltInclude>

Description

The <XmltInclude> is a sub element of the <OutputFiles> with a boolean value. This element can't contain sub elements.

Syntax

```
<XmltInclude>[Value]</XmltInclude>
```

Parameter

Value	True: the xslt file will be added to the outputfiles collection False: the xslt file won't be added to the outputfiles collection Default: False
-------	--

Example 1

The following example adds the xslt file to the collection of files that can be used in the <DestinationSets> (outputfiles). If that is done, you the file can be saved in the destinationsets to a destination.

```
Function:    <OutputFiles>
             <XmltInclude>True</XmltInclude>
             </OutputFiles>
```

Example 2

The following example doesn't add the generated Xml file to the collection of outputfiles. If that is done, the file cannot be saved in the destinationsets to a destination.

```
Function:    <OutputFiles>
             <XmltInclude>False</XmltInclude>
             </OutputFiles>
```

<Rendering>

Description

The <Rendering> element can contain sub elements. In the type attribute you can define the renderer type by keyword. The sub elements depend on the chosen type.

Syntax

```
<Rendering Type = "<Value>" >
</Rendering>
```

Parameter

Value	The type defines the rendering engine by keyword Available types: EdcServer
-------	--

Example

In the following example the generated xml file will be sent for rendering to a EDC Server with the specified url.

```
Function:    <Logging>
             <Rendering Type ="EdcServer" >
               <Url>http://ourserver.net/officeatwork/edcserver</Url>
             </Rendering>
             </Logging>
```

<PollForDownload>

Description

The <PollForDownload> element is an optional sub element of the <Rendering> element, type “EdcServer”. The value is a boolean value to control the polling for the document download from the EDC Server.

Syntax

```
<PollForDownload>[Value]</PollForDownload>
```

Parameter

Value	Boolean value to control the polling for document download Default: TRUE
-------	---

Example

The following example sends the generated xml file to the EDC-Server instance with url http://ourserver.net/officeatwork/edcserver for rendering the documents, but skips waiting for the generated document of the EDC Server.

```
Function:    <Rendering Type ="EdcServer">
             <Url>http://ourserver.net/officeatwork/edcserver</Url>
             <PollForDownload>False</PollForDownload>
             </Rendering>
```

<Url>

Description

The <Url> element is a sub element of the <Rendering> element, type “EdcServer”. The value is stringed with the url to the EDC Server instance. This element can’t contain sub elements.

Syntax

```
<Url>[Value]</Url>
```

Parameter

Value	Url to the EDC Server instance
-------	--------------------------------

Example

The following example sends the generated xml file to the EDC Server instance with url http://ourserver.net/officeatwork/edcserver for rendering the documents.


```
Function:    <Rendering Type ="EdcServer">
             <Url>http://ourserver.net/officeatwork/edcserver</Url>
             </Rendering>
```

<DestinationSets>

Description

The <DestinationSets> element is a sub element of the <Logging> element and contains sub elements of type <DestinationSet>.

Syntax

```
<DestinationSets>
</DestinationSets>
```

Example

In the following example one DestinationSet is defined. For all files of the outputfiles collection an annotation with a relationship to an existing contact entity will be created.

```
Function:    <Logging>
             <DestinationSets>
               <DestinationSet>
                 <Filters>
                   <Filter Type="FileName">*. *</Filter>
                 </Filters>
                 <Destinations>
                   <Destination Type="DynamicsCrmNote">
                     <Relationships>
                       <Relationship>
                         <Entity>contact</Entity>
                         <Guid>{{Field('contactid')}}</Guid>
                       </Relationship>
                     </Relationships>
                   </Destination>
                 </Destinations>
               </DestinationSet>
             </DestinationSets>
           </Logging>
```

<DestinationSet>

Description

The <DestinationSet> element is a sub element of <DestinationSets> and contains a sub element <Filters> and a sub element <Destinations>.

Syntax

```
<DestinationSet>
</DestinationSet>
```

Example

In the following example one DestinationSet is defined. For all files in the outputfiles collection an annotation with a relationship to an existing contact entity will be created.

```
Function:    <DestinationSets>
             <DestinationSet>
               <Filters>
                 <Filter Type="FileName">*. *</Filter>
```

```
        </Filters>
        <Destinations>
            <Destination Type="DynamicsCrmNote">
                <Relationships>
                    <Relationship>
                        <Entity>contact</Entity>
                        <Guid>{{Field('contactid')}}</Guid>
                    </Relationship>
                </Relationships>
            </Destination>
        </Destinations>
    </DestinationSet>
</DestinationSets>
```

<Filters>

Description

The <Filters> element is a sub element of <DestinationSet> and can contain sub elements of type <Filter>.

Syntax

```
<Filters>
</Filters>
```

Example

In the following example one <Filters> element is defined. The files in the outputfiles collection will be filtered by name.

```
Function:    <DestinationSet>
             <Filters>
               <Filter Type="FileName">*.*</Filter>
             </Filters>
           </DestinationSet>
```

<Filter>

Description

The <Filter> element is a sub element of <Filters> and can't contain sub elements. The <Filter> element is defined by a type attribute.

The <Filter> definitions are additive. If one file satisfies more than one filter criteria it will be added only once to the output.

Syntax

```
<Filter Type="[Type]">[Value]</Filter>
```

Parameter

Type	Defines the filter type Available filter types: FileName
Value	The value definition depends on the filter type Definition for available filter types: FileName: String with wildcards *

Example 1

In the following example all files from the outputfiles collection will be proceeded.

```
Function:    <Filters>
            <Filter Type="FileName">*.*/Filter>
            </Filters>
```

Example 2

In the following example only files from the outputfiles collection with the extension "docx" will be proceeded.

```
Function:    <Filters>
            <Filter Type="FileName">*.docx</Filter>
            </Filters>
```

Example 3

In the following example only files from the outputfiles collection with the extension "docx" or the extension "pdf" will be proceeded.

```
Function:    <Filters>
            <Filter Type="FileName">*.docx</Filter>
            <Filter Type="FileName">*.pdf</Filter>
            </Filters>
```

Example 4

In the following example only files from the outputfiles collection with part of the filename "customer" and the extension "docx" will be proceeded.

```
Function:    <Filters>
            <Filter Type="FileName">*customer*.docx</Filter>
            </Filters>
```

<Destinations>

Description

The <Destinations> element is a sub element of the <DestinationSet> and can contain sub elements of type <Destination>.

Syntax

```
<Destinations>
</Destinations>
```

Example

In the following example is one <Destination> element of type "DynamicsCrmNote" in the <Destinations> element defined.

```
Function:    <DestinationSet>
            <Destinations>
                <Destination Type="DynamicsCrmNote">
                    <Relationships>
                        <Relationship>
                            <Entity>contact</Entity>
                            <Guid>{{Field('contactid')}}</Guid>
                        </Relationship>
                    </Relationships>
                </Destination>
            </Destinations>
        </DestinationSet>
```

<Destination>

Description

The <Destination> element is a sub element of the <Destinations> element and can contain sub elements. The element type is defined by the type attribute.

The sub elements of the <Destination> depend on the destination type.

Syntax

```
<Destination Type="[Type]">
</Destination>
```

Parameter

Type	Defines the destination type Available destination types: DynamicsCrmNote, FilePath
------	--

Example 1

In the following example is one <Destination> element of type "DynamicsCrmNote" in the <Destinations> element defined.

```
Function:    <Destinations>
              <Destination Type="DynamicsCrmNote">
                <Relationships>
                  <Relationship>
                    <Entity>contact</Entity>
                    <Guid>{{Field('contactid')}}</Guid>
                  </Relationship>
                </Relationships>
              </Destination>
            </Destinations>
```

Example 2

In the following example are two <Destination> elements of type "DynamicsCrmNote" in the <Destinations> element defined.

```
Function:    <Destinations>
              <Destination Type="DynamicsCrmNote">
                <Relationships>
                  <Relationship>
                    <Entity>contact</Entity>
                    <Guid>{{Field('contactid')}}</Guid>
                  </Relationship>
                </Relationships>
              </Destination>
              <Destination Type="DynamicsCrmNote">
                <Relationships>
                  <Relationship>
                    <Entity>account</Entity>
                    <Guid>{{Field('accountid')}}</Guid>
                  </Relationship>
                </Relationships>
              </Destination>
            </Destinations>
```

Example 3

In the following example are two <Destination> elements defined. One of type "DynamicsCrmNote" and one of type "FilePath".

```
Function:    <Destinations>
              <Destination Type="DynamicsCrmNote">
                <Relationships>
                  <Relationship>
                    <Entity>contact</Entity>
                    <Guid>{{Field('contactid')}}</Guid>
```

```

        </Relationship>
    </Relationships>
</Destination>
<Destination Type="FilePath">
    <Path>\\myserver\backups</Path>
    <Overwrite>True</Overwrite>
</Destination>
</Destinations>

```

<Destination Type="DynamicsCrmNote">

Description

The <Destination Type="DynamicsCrmNote"> element is a sub element of the <Destinations> element and can contain a sub element of type <Relationships>. The <Relationships> element can contain <Relationship> elements. Each <Relationship> element has the two sub elements, <Entity> and <Guid>.

Syntax

```

<Destination Type="DynamicsCrmNote">
    <Relationships>
        <Relationship>
            <Entity>[Value]</Entity>
            <Guid>[Value]</Guid>
        </Relationship>
    </Relationships>
</Destination>

```

Parameter

Entity	Name of the entity for the relationship to the annotation
Guid	Guid of the entity for the relationship to the annotation

Example 1

In the following example is one <Destination> element of type "DynamicsCrmNote" with one <Relationship>.

```

Function:    <Destination Type="DynamicsCrmNote">
            <Relationships>
                <Relationship>
                    <Entity>contact</Entity>
                    <Guid>{{Field('contactid')}}</Guid>
                </Relationship>
            </Relationships>
        </Destination>

```

Example 2

In the following example is one <Destination> element of type "DynamicsCrmNote" with two <Relationship>: One annotation to a contact entity and one annotation to an account entity.

```

Function:    <Destination Type="DynamicsCrmNote">
            <Relationships>
                <Relationship>
                    <Entity>contact</Entity>
                    <Guid>{{Field('contactid')}}</Guid>
                </Relationship>
                <Relationship>
                    <Entity>account</Entity>
                    <Guid>{{RelatedField('accountid')}}</Guid>
                </Relationship>
            </Relationships>
        </Destination>

```

<Destination Type="FilePath">

Description

The <Destination Type="FilePath"> element is a sub element of the <Destination> element. It contains the sub element <Path> and can contain the sub element <Overwrite>.

Syntax

```
<Destination Type="FilePath">
  <Path>[Value]</Path>
  <Overwrite>[Value]</Overwrite>
</Destination>
```

Parameter

Path	Path for saving the outputfiles
Overwrite	Optional boolean value to overwrite an existing file True: The existing file will be overwritten False: The existing file persists and the new file won't be saved. Default: True

Example 1

In the following example is one <Destination> element of type "FilePath" with a sub element <Path>. The files will be saved to this path. If there is an existing file, it will be overwritten.

```
Function: <Destination Type="FilePath">
          <Path>D:\Backup\Documents</Path>
        </Destination>
```

Example 2

In the following example is one <Destination> element of type "FilePath" with a sub element <Path>. The files will be saved to this path. If there is an existing file, it won't be overwritten.

```
Function: <Destination Type="FilePath">
          <Path>D:\Backup\Documents</Path>
          <Overwrite>False</Overwrite>
        </Destination>
```

Example 3

In the following example are two <Destination> elements of type "FilePath". The outputfile will be saved twice. One time the files are saved to the local device as backup. If there is an existing file it will be overwritten. Second time the files are saved to a network drive for printing and sending a hard copy to the customer.

```
Function: <Destination Type="FilePath">
          <Path>D:\Backup\Documents</Path>
          <Overwrite>True</Overwrite>
        </Destination>
        <Destination Type="FilePath">
          <Path>\\ourserver\packingqueue</Path>
          <Overwrite>False</Overwrite>
        </Destination>
```

XML Web Resources

Use Cases

You can use the «XML Web Resources» to manage repetitively used information only one time. For better understanding, we explain several use cases in this chapter.

Translations

You can use the XML webresource to centralize some translations like tags and currencies.

Example

In the following example we read the translated value for the swiss currency from the web resource "new_translations" (written in xml format).

```
Function:    {{XmlWebResourceValue('new_translations';  
        '//Translations/Label[@Name="Currency_CHF"]/Translation[@LCID=2055]  
        ')}}
```

Result: Franken

Settings

XML webresource can also be used to centralize settings.

Example

In the following example we read the setting value for the edc webservice from the "new_settings" (written in xml format) in dependence of the client.

```
Function:    {{XmlWebResourceValue('new_settings '  
        '//Settings/Label[@Name="edcserver"]/UrlForCompany[@Client=CompanyX  
        Y]')}}}
```

Result: http://edcserver.companyxy.com/

Appendix

OSCT examples

Start officeatwork Template Chooser

In the following osct file example is a language code (Language) but no template (TemplateID) defined. In case that this file proceeds on the users client, the officeatwork Template Chooser is started and the user has to select the template.

The generated OSC-File will be attached to the notes of the contact where the document generating process was started.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Parameters>

  <CreateDocument>
    <Language>2055</Language>
    <TemplateID></TemplateID>
  </CreateDocument>

  <Processing>
    <Logging>
      <OutputFiles>
        <OscInclude>True</OscInclude>
        <OsctInclude>False</OsctInclude>
      </OutputFiles>
    </Logging>

    <DestinationSets>
      <DestinationSet>
        <Filters>
          <Filter Type="FileName">*.osc</Filter>
        </Filters>
        <Destinations>
          <Destination Type="DynamicsCrmNote">
            <Relationships>
              <Relationship>
                <Entity>contact</Entity>
                <Guid>{{Field('contactid')}}</Guid>
              </Relationship>
            </Relationships>
          </Destination>
        </Destinations>
      </DestinationSet>
    </DestinationSets>
  </Processing>

</Parameters>

<!------->
<!--Copyright officeatwork AG, Switzerland-->
<!------->
```

Create Letter to Contact

The following osct file is a simple example to generate a letter in Swiss German to a contact. Various values are read from the contact entity.

The generated OSC-File will be attached to the notes of the contact where the document generating process was started.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Parameters>
  <CreateDocument>

    <TemplateID>Letter</TemplateID>
    <Language>2055</Language>
    <ShowDocumentWizard>-1</ShowDocumentWizard>

    <MasterProperties>
      <MasterProperty IDName="Recipient">
        <Field Name="FirstName" Value="{{Field('firstname')}}" />
        <Field Name="LastName" Value="{{Field('lastname')}}" />
        <Field Name="FullName" Value="{{Field('firstname')}} {{Field('lastname')}}" />
        <Field Name="CompleteAddress" Value="{{Field('firstname')}} {{Field('lastname')}}
{{Field('address1_line1')}}
{{Field('address1_postalcode')}} {{Field('address1_city')}}
{{Field('address1_country')}}" />
        <Field Name="Introduction" Value="{{Field('salutation')}} {{Field('lastname')}}" />
        <Field Name="EMail" Value="{{Field('emailaddress1')}}" />
      </MasterProperty>
      <MasterProperty IDName="Organisation" Where="IDName" Is="Contoso Ltd, Zürich, Headquarters" />
      <MasterProperty IDName="Author" Where="Name" Is="Peter Muster" />
      <MasterProperty IDName="Contactperson" Where="Name" Is="Peter Muster" />
      <MasterProperty IDName="Signature1" Where="Name" Is="Peter Muster" />
      <MasterProperty IDName="Signature2" Where="Name" Is="Cyrill Hutter" />
      <MasterProperty IDName="CustomField">
        <Field Name="DocumentDate" Value="{{Today('dd.MM.yyyy')}}" />
        <Field Name="DocumentSubject" Value="Letter to {{Field('salutation')}} {{Field('firstname')}}
{{Field('lastname')}}" />
      </MasterProperty>
    </MasterProperties>

    <Output>
      <Save UID="2006121210441235887611" ShowDialog="0" Path="%Desktop%" Filename="Letter to Contact
{{Field('firstname')}} {{Field('lastname')}}.docx" ReplaceExistingFile="-1" />
    </Output>

  </CreateDocument>

</Processing>

<Logging>
  <OutputFiles>
    <OscInclude>True</OscInclude>
    <OsctInclude>False</OsctInclude>
  </OutputFiles>
</Logging>

<DestinationSets>
  <DestinationSet>
    <Filters>
      <Filter Type="FileName">*.osct</Filter>
    </Filters>
    <Destinations>
      <Destination Type="DynamicsCrmNote">
        <Relationships>
          <Relationship>
            <Entity>contact</Entity>
            <Guid>{{Field('contactid')}}</Guid>
          </Relationship>
        </Relationships>
      </Destination>
    </Destinations>
  </DestinationSet>
</DestinationSets>

</Processing>

</Parameters>
```

Create Letter to primary Contact of Account

The following osct file is a simple example to generate a letter in English US to the primary contact of an account. Various values are read from the account entity using the `{{Field()}}`-Function and additional values are read from the contact entity related by the `primarycontactid` field using the `{{RelatedField()}}`-Function.

The generated OSC-File will be attached to the notes of the account where the document generating process was started and the primary contact.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<Parameters>
  <CreateDocument>
    <TemplateID>Letter</TemplateID>
    <ShowDocumentWizard>-1</ShowDocumentWizard>
    <Language>1033</Language>
    <MasterProperties>
      <MasterProperty IDName="Recipient">
        <Field Name="FirstName" Value="{{RelatedField('primarycontactid'; 'contact'; 'firstname')}}"
      />
      <Field Name="LastName" Value="{{RelatedField('primarycontactid'; 'contact'; 'lastname')}}" />
      <Field Name="FullName" Value="{{RelatedField('primarycontactid'; 'contact'; 'firstname')}}
      {{RelatedField('primarycontactid'; 'contact'; 'lastname')}}" />
      <Field Name="CompleteAddress" Value="{{Field('name')}}
      {{RelatedField('primarycontactid'; 'contact'; 'firstname')}} {{RelatedField('primarycontactid'; 'contact';
      'lastname')}}
      {{RelatedField('primarycontactid'; 'contact'; 'address1_line1')}}
      {{RelatedField('primarycontactid'; 'contact'; 'address1_postalcode')}} {{RelatedField('primarycontactid';
      'contact'; 'address1_city')}}
      {{RelatedField('primarycontactid'; 'contact'; 'address1_country')}}" />
      <Field Name="Introduction" Value="{{RelatedField('primarycontactid'; 'contact';
      'salutation')}} {{RelatedField('primarycontactid'; 'contact'; 'firstname')}}
      {{RelatedField('primarycontactid'; 'contact'; 'lastname')}}" />
    </MasterProperty>
    <MasterProperty IDName="Organisation" Where="IDName" Is="Contoso US Ltd, New York, Marketing"/>
    <MasterProperty IDName="Author" Where="Name" Is="Sandra Iten"/>
    <MasterProperty IDName="Contactperson" Where="Name" Is="Sandra Iten"/>
    <MasterProperty IDName="Signature1" Where="Name" Is="Sandra Iten"/>
    <MasterProperty IDName="Signature2" Where="Name" Is="Andrea Anderegger"/>
    <MasterProperty IDName="CustomField">
      <Field Name="DocumentDate" Value="{{Today('dd.MM.yyyy')}}" />
      <Field Name="DocumentSubject" Value="Letter to primary contact
      {{RelatedField('primarycontactid'; 'contact'; 'firstname')}} {{RelatedField('primarycontactid'; 'contact';
      'lastname')}} of account {{Field('name')}}" />
    </MasterProperty>
  </MasterProperties>
  <Output>
    <Save UID="2006121210441235887611" ShowDialog="0" Path="%Desktop%"
    Filename="{{Today('dd.MM.yyyy')}} Letter to primary contact {{RelatedField('primarycontactid'; 'contact';
    'firstname')}} {{RelatedField('primarycontactid'; 'contact'; 'lastname')}} of account
    {{Field('name')}}.docx" ReplaceExistingFile="-1" />
  </Output>
</CreateDocument>
<Processing>
  <Logging>
    <OutputFiles>
      <OscInclude>True</OscInclude>
      <OscInclude>False</OscInclude>
    </OutputFiles>
  </Logging>
  <DestinationSets>
    <DestinationSet>
      <Filters>
        <Filter Type="FileName">*.osc</Filter>
      </Filters>
      <Destinations>
        <Destination Type="DynamicsCrmNote">
          <Relationships>
            <Relationship>
              <Entity>account</Entity>
              <Guid>{{Field('accountid')}}</Guid>
            </Relationship>
            <Relationship>
              <Entity>contact</Entity>
              <Guid>{{RelatedField('primarycontactid'; 'contact'; 'contactid')}}</Guid>
            </Relationship>
          </Relationships>
        </Destination>
      </Destinations>
    </DestinationSet>
  </DestinationSets>
</Processing>
</Parameters>

```

Create project reports

The following osct file creates a project report, started from the “new_partnercontract” entity. The document language is set in dependence of the account field “new_invoicedocumenlanguage”. Some information (lastname, firstname, ...) about the customer projectmanager and the project leader are read.

In the second part a loop over the related incidents and the change requests is used.

```
<Parameters>
  <CreateDocument>
    <Language>{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="LCID"]/Translation[@LCID="{{RelatedField('new_account'; 'account';
  'new_invoicedocumenlanguage')}}"]')}}</Language>
    <TemplateID>Report_New</TemplateID>
    <ShowDocumentWizard>0</ShowDocumentWizard>

    <MasterProperties>
      <MasterProperty IDName="Recipient">
        <Field Name="IDName" Value="{{RelatedField('new_customerprojectmanager'; 'contact';
  'lastname')}} {{RelatedField('new_customerprojectmanager'; 'contact'; 'firstname')}}
  {{RelatedField('new_account'; 'account'; 'name')}}"/>
        <Field Name="CompleteAddress" Value="{{RelatedField('new_account'; 'account'; 'name')}}
  {{RelatedField('new_customerprojectmanager'; 'contact'; 'salutation')}}
  {{RelatedField('new_customerprojectmanager'; 'contact'; 'firstname')}}
  {{RelatedField('new_customerprojectmanager'; 'contact'; 'lastname')}}
  {{RelatedField('new_account'; 'account'; 'address1_line1')}}
  {{RelatedField('new_account'; 'account'; 'address1_postalcode')}} {{RelatedField('new_account'; 'account';
  'address1_city')}}
  {{RelatedField('new_account'; 'account'; 'address1_country')}}"/>
        <Field Name="Company" Value="{{RelatedField('new_account'; 'account'; 'name')}}"/>
        <Field Name="FullName" Value="{{RelatedField('new_customerprojectmanager'; 'contact';
  'lastname')}} {{RelatedField('new_customerprojectmanager'; 'contact'; 'firstname')}}"/>
        <Field Name="AddressStreet" Value="{{RelatedField('new_account'; 'account';
  'address1_line1')}}"/>
        <Field Name="AddressZIP" Value="{{RelatedField('new_account'; 'account';
  'address1_postalcode')}}"/>
        <Field Name="AdressCity" Value="{{RelatedField('new_account'; 'account'; 'address1_city')}}"/>
        <Field Name="Telephone" Value="{{RelatedField('new_customerprojectmanager'; 'contact';
  'telephonenumber')}}"/>
        <Field Name="EMail" Value="{{RelatedField('new_customerprojectmanager'; 'contact';
  'emailaddress1')}}"/>
      </MasterProperty>
      <MasterProperty IDName="CustomField">
        <Field Name="DocumentType" Value="Projekt Report"/>
        <Field Name="DocumentSubject" Value="{{Field('new_name')}}"/>
        <Field Name="YourReference" Value="{{RelatedField('new_customerprojectmanager'; 'contact';
  'firstname')}} {{RelatedField('new_customerprojectmanager'; 'contact'; 'lastname')}}
  {{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="Global_Project"]/Translation[@LCID="{{RelatedField('new_account'; 'account';
  'new_invoicedocumenlanguage')}}"]')}}: {{Field('new_name')}}
  {{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="Global_Account"]/Translation[@LCID="{{RelatedField('new_account'; 'account';
  'new_invoicedocumenlanguage')}}"]')}}: {{RelatedField('new_account'; 'account'; 'name')}}"/>
      </MasterProperty>
      <MasterProperty IDName="Company" Where="OID" Is="Projects"/>
      <MasterProperty IDName="Contactperson" Where="EID" Is="{{RelatedField('new_projectmanager';
  'systemuser'; 'new_employeeid')}}"/>
      <MasterProperty IDName="Signature1" Where="EID" Is="{{RelatedField('new_projectmanager';
  'systemuser'; 'new_employeeid')}}"/>
    </MasterProperties>

    <Bookmarks>
      <Bookmark Name="Subject" Value="Projekt Report"/>
    </Bookmarks>

    <Contents>
      <Content ID = "PROJ_ProjectReport_010_ProjectOverview">
        <Value Name="Currency" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="CurrencySymbol_{{RelatedField('new_account'; 'account';
  'transactioncurrencyid')}}"])/Translation[@LCID="{{RelatedField('new_account'; 'account';
  'new_invoicedocumenlanguage')}}"]')}}"/>
        <Value Name="ProjectName" Value="{{Field('new_name')}}"/>
        <Value Name="ProjectStart" Value="{{Field('new_startdate'; '')}}"/>
        <Value Name="ProjectPhase" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="ProjectPhase_{{Field('new_projectphase')}}"])/Translation[@LCID="{{RelatedField
  ('new_account'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
        <Value Name="CustomerProjectManager" Value="{{RelatedField('new_customerprojectmanager';
  'contact'; 'firstname')}} {{RelatedField('new_customerprojectmanager'; 'contact'; 'lastname')}}
  {{RelatedField('new_customerprojectmanager'; 'contact'; 'telephonenumber')}}
  {{RelatedField('new_customerprojectmanager'; 'contact'; 'emailaddress1')}}"/>
      </Content>
    </Contents>
  </CreateDocument>
</Parameters>
```

```

    <Value Name="officeatworkProjectManager" Value="{{RelatedField('new_projectmanager';
'systemuser'; 'firstname')}} {{RelatedField('new_projectmanager'; 'systemuser'; 'lastname')}}
{{RelatedField('new_projectmanager'; 'systemuser'; 'address1_telephone1')}}
{{RelatedField('new_projectmanager'; 'systemuser'; 'internalemailaddress')}}"/>
    <Value Name="officeatworkProjectManagerDeputy"
Value="{{RelatedField('new_deputyprojectmanager'; 'systemuser'; 'firstname')}}
{{RelatedField('new_deputyprojectmanager'; 'systemuser'; 'lastname')}}
{{RelatedField('new_deputyprojectmanager'; 'systemuser'; 'address1_telephone1')}}
{{RelatedField('new_deputyprojectmanager'; 'systemuser'; 'internalemailaddress')}}"/>
    <Value Name="SumOffered" Value="{{Field('new_offer')}}"/>
    <Value Name="SumChangeRequests" Value="{{Field('new_changerequests')}}"/>
    <Value Name="SumToBeInvoiced" Value="{{Field('new_tobeinvoiced')}}"/>
    <Value Name="SumInvoiced" Value="{{Field('new_invoiced')}}"/>
    <Value Name="SumBalance" Value="{{Field('new_balance')}}"/>
</Content>

{{If('{{RelatedCount('new_projectincident'; 'new_project')}}>0';
'<Content ID = "PROJ_ProjectReport_100_TitleIncidentTypes"/>
{{RelatedLoop('IncidentLoop'; 'new_projectincident'; 'new_project';
'<Content ID = "PROJ_ProjectReport_110_BodyIncidentTypes">
<Value Name="IncidentType" Value = "{{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="IncidentType {{RelatedLoopField('IncidentLoop';
'new_incidenttype')}}]/Translation[@LCID="{{RelatedField('new_account'; 'account';
'new_invoicedocumenlanguage')}}"]')}}"/>
<Value Name="Offered" Value = "{{RelatedLoopField('IncidentLoop'; 'new_offer')}}"/>
<Value Name="ChangeRequests" Value = "{{RelatedLoopField('IncidentLoop';
'new_changerequests')}}"/>
<Value Name="ToBeInvoiced" Value = "{{RelatedLoopField('IncidentLoop';
'new_tobeinvoiced')}}"/>
<Value Name="Invoiced" Value = "{{RelatedLoopField('IncidentLoop'; 'new_in')}}"/>
<Value Name="Balance" Value = "{{RelatedLoopField('IncidentLoop'; 'new_balance')}}"/>
<Value Name="Currency" Value="{{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="CurrencySymbol_{{RelatedField('new_account'; 'account';
'transactioncurrencyid')}}]/Translation[@LCID="{{RelatedField('new_account'; 'account';
'new_invoicedocumenlanguage')}}"]')}}"/>
</Content>')}}

<Content ID = "PROJ_ProjectReport_120_TotalIncidentTypes">
<Value Name="SumBalance" Value="{{Field('new_balance')}}"/>
<Value Name="Currency" Value="{{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="CurrencySymbol_{{RelatedField('new_account'; 'account';
'transactioncurrencyid')}}]/Translation[@LCID="{{RelatedField('new_account'; 'account';
'new_invoicedocumenlanguage')}}"]')}}"/>
</Content>
')}}

{{If('{{RelatedCount('new_projectchangerequest'; 'new_project')}}>0';
'<Content ID = "PROJ_ProjectReport_200_TitleChangeRequests"/>
{{RelatedLoop('ChangeRequestLoop'; 'new_projectchangerequest'; 'new_project'; 'new_name';
'true';
'<Content ID = "PROJ_ProjectReport_210_BodyChangeRequests">
<Value Name="IncidentType" Value = "{{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="IncidentType_{{RelatedLoopField('ChangeRequestLoop';
'new_incidenttype')}}]/Translation[@LCID="{{RelatedField('new_account'; 'account';
'new_invoicedocumenlanguage')}}"]')}}"/>
<Value Name="Name" Value = "{{RelatedLoopField('ChangeRequestLoop'; 'new_name')}}"/>
<Value Name="Currency" Value="{{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="CurrencySymbol_{{RelatedField('new_account'; 'account';
'transactioncurrencyid')}}]/Translation[@LCID="{{RelatedField('new_account'; 'account';
'new_invoicedocumenlanguage')}}"]')}}"/>
<Value Name="Price" Value = "{{RelatedLoopField('ChangeRequestLoop'; 'new_price')}}"/>
</Content>')}}

<Content ID = "PROJ_ProjectReport_220_TotalChangeRequests">
<Value Name="SumChangeRequests" Value="{{Field('new_changerequests')}}"/>
<Value Name="Currency" Value="{{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="CurrencySymbol_{{RelatedField('new_account'; 'account';
'transactioncurrencyid')}}]/Translation[@LCID="{{RelatedField('new_account'; 'account';
'new_invoicedocumenlanguage')}}"]')}}"/>
</Content>
')}}
</Contents>

</CreateDocument>
</Parameters>

<!--+++++----->
<!--Copyright officeatwork AG, Switzerland-->
<!--+++++----->

```

Generate partner contract

In the following osct file the document Language Id is read from the account entity and translated into a Microsoft LCID.

In the Masterproperties some information about the related owner (systemuser) of the partner contract are read. Also his Employee Id is used to set his personal informations in the document and put in his signature.

Then we use different Smart-Contents to build the document, the partner type is read from account.

In the output section the filename is built dynamically for the output file in dependence of the partner type of the partner contract.

```
<Parameters>
  <CreateDocument>

    <TemplateID>Contract</TemplateID>
    <ShowDocumentWizard>0</ShowDocumentWizard>
    <Language>{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="LCID"]/Translation[@LCID="{{RelatedField('new_account'; 'account';
  'new_invoicedocumenlanguage')}}"]')}}</Language>

    <MasterProperties>
      <MasterProperty IDName="CustomField">
        <Field Name="DocumentType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="Agreement"]/Translation[@LCID="{{RelatedField('new_account'; 'account';
  'new_invoicedocumenlanguage')}}"]')}}"/>
        <Field Name="DocumentSubject" Value="officeatwork {{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
        <Field Name="DocumentDate" Value="{{Field('new_partnercontractdate'; '')}}"/>
      </MasterProperty>
      <MasterProperty IDName="Company" Where="OID" Is="Contracts"/>
      <MasterProperty IDName="Contactperson" Where="EID" Is="{{RelatedField('ownerid'; 'systemuser';
  'new_employeeid')}}"/>
      <MasterProperty IDName="Signature1" Where="EID" Is="{{RelatedField('ownerid'; 'systemuser';
  'new_employeeid')}}"/>
    </MasterProperties>

    <Contents>
      <Content ID = "LEGL_PartnerContract_000 Header">
        <Value Name="PartnerType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
        <Value Name="PartnerCompleteAddress" Value="{{Field('new_completeaddress')}}"/>
      </Content>
      <Content ID = "LEGL_PartnerContract_120 Präambel Ambassador">
        <Value Name="PartnerType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
      </Content>
      <Content ID = "LEGL_PartnerContract_180 Vertragsgegenstand Ambassador">
        <Value Name="PartnerType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
      </Content>
      <Content ID = "LEGL_PartnerContract_200 Vertragsgebiet">
        <Value Name="PartnerType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
      </Content>
      <Content ID = "LEGL_PartnerContract_300 Ausschluss der Exklusivität">
        <Value Name="PartnerType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
      </Content>
      <Content ID = "LEGL_PartnerContract_407 Rechte und Pflichten des officeatwork Ambassadors">
        <Value Name="PartnerType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
      </Content>
      <Content ID = "LEGL_PartnerContract_408 Vermittlungsrecht officeatwork Ambassadors">
        <Value Name="PartnerType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
      </Content>
      <Content ID = "LEGL_PartnerContract_409 Anforderungen officeatwork Ambassador">
        <Value Name="PartnerType" Value="{{XmlWebResourceValue('new_translations';
  //Translations/Label[@Name="PartnerType_{{Field('new_type')}}]/Translation[@LCID="{{RelatedField('new_acc
  ount'; 'account'; 'new_invoicedocumenlanguage')}}"]')}}"/>
      </Content>
    </Contents>
  </CreateDocument>
</Parameters>
```

```

    <Content ID = "LEGL_PartnerContract_410 Sorgfaltspflicht">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_437 Unteragentur Ambassador">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_440 Konkurrenzverbot">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_450 Vertraulichkeit">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_460 Feedback">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_480 Akquisitions-Kosten">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_600 Rechte und Pflichten von officeatwork">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_610 Unterstützung">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_620 Partner-Betreuer">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_630 Marketingmaterial">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_640 officeatwork Connect">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_680 Lead Schutz Ambassador">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_710 Provisionsanspruch Ambassador">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_920 Geistiges Eigentum Ambassador">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_1100 Vertragsdauer und Beendigung">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_1200 Abschluss">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_1300 Kündigung">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>
    <Content ID = "LEGL_PartnerContract_1400 Kündigung aus wichtigem Grund">
      <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
    </Content>

```



```

        <Content ID = "LEGL_PartnerContract_1510 Folgen der Vertragsauflösung Ambassador">
        <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
        </Content>
        <Content ID = "LEGL_PartnerContract_1800 Haftung">
        <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
        </Content>
        <Content ID = "LEGL_PartnerContract_2200 Anstellungsverzicht">
        <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
        </Content>
        <Content ID = "LEGL_PartnerContract_5100 Geheimhaltung, Datenschutz">
        <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
        </Content>
        <Content ID = "LEGL_PartnerContract_6000 Allgemeines">
        <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
        </Content>
        <Content ID = "LEGL_PartnerContract_9000 Signatures">
        <Value Name="PartnerType" Value="{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}"/>
        <Value Name="officeatworkSignee" Value="{{RelatedField('ownerid'; 'systemuser'; 'firstname')}}
{{RelatedField('ownerid'; 'systemuser'; 'lastname')}}"/>
        <Value Name="PartnerLegalEntity" Value="{{RelatedField('new_account'; 'account'; 'name')}}"/>
        <Value Name="PartnerCity" Value="{{Field('new_city')}}"/>
        <Value Name="PartnerSignee" Value="{{RelatedField('new_partnercontact'; 'contact';
'firstname')}} {{RelatedField('new_partnercontact'; 'contact'; 'lastname')}}"/>
        <Value Name="ContractStartDate" Value="{{Field('new_startdate')}}"/>
        </Content>
    </Contents>

    <Output>
        <Save Filename="Agreement {{XmlWebResourceValue('new_translations';
'//Translations/Label[@Name="PartnerType_{{Field('new_type')}}']/Translation[@LCID="{{RelatedField('new_acc
ount'; 'account'; 'new_invoicedocumenlanguage')}}"])}" FileFormat="DOCX" DeliveryMethod="Download"
/>
    </Output>

</CreateDocument>
</Parameters>

<!--+++++----->
<!--Copyright officeatwork AG, Switzerland-->
<!--+++++----->

```

Generate mass output for sales special

In the following DMCLT file a letter, based on the Smart-Template “Letter Christmas Special”, will be generated for each account.

```
<DCML>
  <Instruction>

  </Instruction>

  <Data>
    <Processing>
      <Logging>
        <OutputFiles>
          <XmlInclude>True</XmlInclude>
          <XsltInclude>True</XsltInclude>
        </OutputFiles>
      </Logging>

      <Rendering Type ="EdcServer">
        <Url>http://ourserver/officeatwork/contosocom/contosocom/edcserver/</Url>
      </Rendering>

      <DestinationSets>
        <DestinationSet>
          <Filters>
            <Filter Type="FileName">*.dcmlt</Filter>
            <Filter Type="FileName">*.docx</Filter>
          </Filters>
          <Destinations>
            <Destination Type="DynamicsCrmNote">
              <Relationships>
                <Relationship>
                  <Entity>offatwrk_job</Entity>
                  <Guid>571A3A24-5AC6-E411-89B7-D89D6763FF78</Guid>
                </Relationship>
              </Relationships>
            </Destination>
          </Destinations>
        </DestinationSet>
      </DestinationSets>
    </Processing>

    <UserData>
      <IdName>Müller Heinz, HM, Entwicklungsleiter</IdName>
      <Department>Accounting</Department>
    </UserData>

    <CRMDData>
      {{FetchXml('<fetch version="1.0" output-format="xml-platform" mapping="logical" distinct="false">
<entity name="account">
  <attribute name="name" />
  <attribute name="address " />
  <attribute name="salutation" />
  <attribute name="closing" />
  <attribute name="languageid" />
  <order attribute="name" descending="false" />
  <filter type="and">
    <condition attribute="statecode" operator="eq" value="0" />
  </filter>
</entity>
</fetch>')}}
    </CRMDData>
  </Data>

</DCML>

<!------->
<!--Copyright officeatwork AG, Switzerland-->
<!------->
```


CHAPTER 6

Support

Get access to a wide range of support resources on officeatwork Connect (connect.officeatwork.com) such as:

- Knowledge Base
- Q & A
- Download Center
- Installers
- Manuals
- Video guides
- Forum
- Glossary
- etc.

To access officeatwork Connect you need to register your Microsoft-Account at www.officeatwork.com → [Connect](#)

All support options and resources can be found on the website www.officeatwork.com → [Support](#)

More services offered by officeatwork such as Education and Consulting can be found on the website www.officeatwork.com → [Services](#)

Index

—B—

Benefit, 6

—C—

Configuration, 8

—D—

Destination, 36

Destination Type

 DynamicsCrmNote, 37

 FilePath, 38

Destinations, 35

DestinationSet, 33

DestinationSets, 33

—F—

FetchXml, 14

FetchXmlCount, 15

FetchXmlLoop, 15

FetchXmlLoopAvg, 18

FetchXmlLoopField, 16

FetchXmlLoopSum, 17

Filter, 34

Filters, 34

Functions, 13

 Field, 13

—I—

If(), 21

Import Solutions, 7

Installation, 7

—J—

JobField, 19

JobLoop, 20

JobLoopField, 21

—L—

Logging, 29

—M—

Menu integration, 9

—N—

Now(), 23

—O—

OutputFiles, 30

Overview, 6

—P—

PollForDownload, 32

Procedure, 11

Processing, 28

—R—

RelatedCount, 24

RelatedField, 23

RelatedLoop, 25

RelatedLoopField, 26

Rendering, 31

—S—

Security, 7

Syntax, 13

Systems, 6

—T—

Today(), 22

Typographic codes & conventions, 5

—U—

Url, 32

—X—

XML structure, 9

XML Template, 11

XML Web Resources, 39

XmlInclude, 30

XsltInclude, 31

XmlWebResourceValue, 27

officeatwork AG
Bundesplatz 12
6300 Zug, Switzerland

T +41 41 544 7100

www.officeatwork.com
mail@officeatwork.com