

# Nuclet: E-Mail-Versand

## Releases

Version	Datum	Hinweise	Kompatibilität
1.1.0	13.02.2018	Erste veröffentlichte Fassung	ab Nuclos 4.25.1
1.1.2	04.07.2019	E-Mail Queue und verbesserte API	ab Nuclos 4.33.8
1.1.3	17.12.2021	E-Mail-Templates	ab Nuclos 4.47.3
1.1.4	22.11.2022	Textbausteine: Unterformularaten allgemein und über Regelbaustein, Versandbedingung für Text- und Anhangsbausteine	ab Nuclos 4.47.3
1.1.5	10.02.2023	Emails als Datei ablegen	ab Nuclos 4.47.3
1.1.6	04.05.2023	wiederkehrende Email-Erinnerungen	ab Nuclos 4.47.3

## Überblick

Das Nuclet "E-Mail-Versand" bietet zusätzlichen Komfort bei der Nutzung der Nuclos Mail API (siehe [http://api.nuclos.de/org/nuclos/api/provider/MailProvider.html#send\(org.nuclos.api.mail.NuclosMail\)](http://api.nuclos.de/org/nuclos/api/provider/MailProvider.html#send(org.nuclos.api.mail.NuclosMail))).

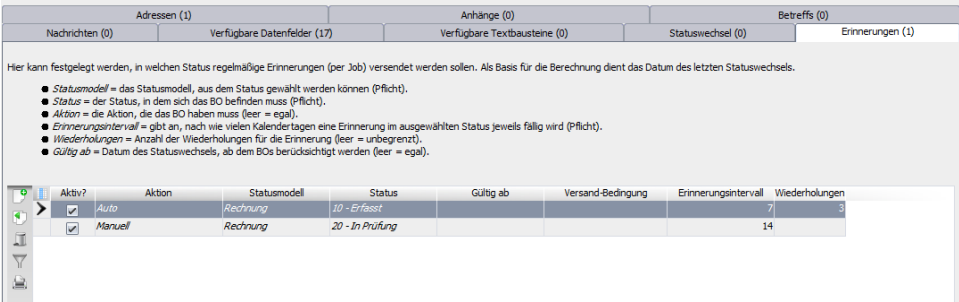
Es ermöglicht insbesondere die Umleitung aller E-Mails zu Testzwecken oder auf Testumgebungen sowie die Archivierung aller versandten E-Mails inkl. Dateianhänge zu Dokumentationszwecken.

Außerdem gibt es einen Job zum Versenden und Empfangen von E-Mails. Damit werden E-Mail etwas später versendet, können nach Fehlern aber zurück gerollt werden, so dass sie nicht mehrfach versendet werden.

Des weiteren gibt es auch die Möglichkeit, Vorlagen / Templates für E-Mails zu konfigurieren.

## Version 1.1.6

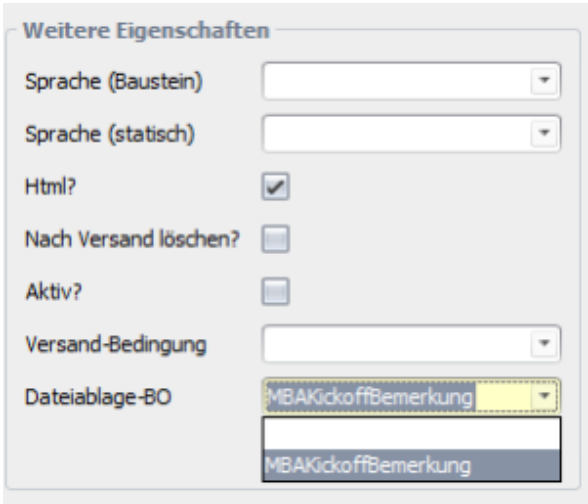
### Beschreibung der Änderungen in Version 1.1.6

Thema	Beschreibung
wiederkehren de Email-Erinnerungen	<p>Der Reiter <i>Erinnerungen</i> im Reiter <i>Konfiguration</i> im <i>E-Mail-Template</i> ermöglicht die Konfiguration wiederkehrender Erinnerungen per E-Mail. Dabei kann für jeden Status ein positiver Wert für das Erinnerungsintervall (<u>Kalendertage</u>) angegeben werden und optional die positive Anzahl an Wiederholungen (ansonsten unbegrenzt).</p> <p>Die Emails werden über den Job <i>Versende Erinnerungen</i> täglich verschickt, falls die Kriterien zutreffen.</p> <p>Erinnerungen können erst ab einem Datum greifen (<i>Gültig ab</i>) und bei Bedarf weiter über eine <i>Versandbedingung</i> eingeschränkt werden.</p>  <p>Dabei ist das Bezugsdatum immer der Tag des relevanten Statuswechsels.</p>

Parameter in E-Mails	<p>Ggf. über Regeln bereitgestellte Parameter (<code>org.nuclet.emailversand.EMailContext.TemplateParameters</code>) können nun auch im Template angesprochen werden, z.B. für die Erinnerungen mit folgender Syntax.</p> <pre> ... die o.g. Rechnung ist seit \$\$ {INTDAYSPAST} Tagen im Status &lt;i&gt;20 - Prüfung&lt;/i&gt;. ... </pre>
----------------------	---

## Version 1.1.5

### Beschreibung der Änderungen in Version 1.1.5

Thema	Beschreibung
Emails als Datei ablegen	<p>Unter <i>Dateiablage-BO</i> im <i>E-Mail-Template</i> werden nun relevante Untertabellen mit Dateianhang zur Auswahl angeboten.</p>  <p>Bei erfolgter Auswahl werden die über dieses Template versendeten Emails im entsprechenden BO als .eml-Datei gespeichert.</p>

## Version 1.1.4

### Beschreibung der Änderungen in Version 1.1.4

Thema	Beschreibung
-------	--------------

Textbausteine: Unterformulardaten allgemein und über Regelbaustein

Man kann nun entweder alle Unterformulardaten in tabellarischer Form ausgeben (Basis-BO generiert aus Unterformular) oder diese zusätzlich über einen Regelbaustein (generiert aus Regelbaustein) einschränken.

E-Mail-Template-Baustein (Textbaustein)

Name:

Baustein-Typ:  Versand-Bedingung:

Aktiv?

E-Mail-Adresse?  Dieser Baustein ist damit im E-Mail-Template auch als Adresse verfügbar.

Sprache?  Dieser Baustein ist damit im E-Mail-Template auch als Sprache verfügbar.

Footer?  Dieser Baustein ist damit im E-Mail-Template auch als Footer verfügbar.

Signatur?  Dieser Baustein ist damit im E-Mail-Template auch als Signatur verfügbar.

1. Basis-BO generiert aus Regelbaustein:

2. Basis-BO generiert aus Unterformular: Nudet:  BO:

Text (0) Verfügbar für (1)

Hier können ebenfalls Datenfelder verwendet werden, wenn sie in dem Basis-BO im eingesetzten Template bzw. unter 1. oder 2. definiertem BO existieren. Endet ein Datenfeld auf "\*", dann wird der Baustein nur dann geschrieben, wenn das gekennzeichnete Datenfeld nicht leer ist.

Text (default)

```
<tr><td>#{Pos}</td><td>#{Standortacalc}</td><td>#{Standortbcalc}</td></tr>
```

E-Mail-Template-Baustein (Textbaustein)

Name:

Baustein-Typ:  Versand-Bedingung:

Aktiv?

E-Mail-Adresse?  Dieser Baustein ist damit im E-Mail-Template auch als Adresse verfügbar.

Sprache?  Dieser Baustein ist damit im E-Mail-Template auch als Sprache verfügbar.

Footer?  Dieser Baustein ist damit im E-Mail-Template auch als Footer verfügbar.

Signatur?  Dieser Baustein ist damit im E-Mail-Template auch als Signatur verfügbar.

1. Basis-BO generiert aus Regelbaustein:

2. Basis-BO generiert aus Unterformular: Nudet:  BO:

Text (0) Verfügbar für (0)

Hier können ebenfalls Datenfelder verwendet werden, wenn sie in dem Basis-BO im eingesetzten Template bzw. unter 1. oder 2. definiertem BO existieren. Endet ein Datenfeld auf "\*", dann wird der Baustein nur dann geschrieben, wenn das gekennzeichnete Datenfeld nicht leer ist.

Text (default)

```
<tr><td>#{Pos}</td><td>#{Standortacalc}</td><td>#{Standortbcalc}</td></tr>
```

Solche Textbausteine werden über einen weiteren Textbaustein als Tabelle ausgegeben.

E-Mail-Template-Baustein (Textbaustein)

Name:

Baustein-Typ:  Versand-Bedingung:

Aktiv?

E-Mail-Adresse?  Dieser Baustein ist damit im E-Mail-Template auch als Adresse verfügbar.

Sprache?  Dieser Baustein ist damit im E-Mail-Template auch als Sprache verfügbar.

Footer?  Dieser Baustein ist damit im E-Mail-Template auch als Footer verfügbar.

Signatur?  Dieser Baustein ist damit im E-Mail-Template auch als Signatur verfügbar.

1. Basis-BO generiert aus Regelbaustein:

2. Basis-BO generiert aus Unterformular: Nudet:  BO:

Text (0) Verfügbar für (1)

Hier können ebenfalls Datenfelder verwendet werden, wenn sie in dem Basis-BO im eingesetzten Template bzw. unter 1. oder 2. definiertem BO existieren. Endet ein Datenfeld auf "\*", dann wird der Baustein nur dann geschrieben, wenn das gekennzeichnete Datenfeld nicht leer ist.

Text (default)

```
<table>
<tr><th>Pos</th><th>Standort A</th><th>Standort B</th></tr>
##{Angebotsposition (Tabellenzeile)}
</table>
```

Versandbedingung für Text- und Anhangsbausteine

Für Text- und Anhangsbausteine kann nun eine Versandbedingung angegeben werden.

**E-Mail-Template-Baustein (Textbaustein)**

Name:

Baustein-Typ:  Versand-Bedingung:

Aktiv?

E-Mail-Adresse?  Dieser Baustein ist damit im E-Mail-Template auch als Adresse verfügbar.

Sprache?  Dieser Baustein ist damit im E-Mail-Template auch als Sprache verfügbar.

Footer?  Dieser Baustein ist damit im E-Mail-Template auch als Footer verfügbar.

Signatur?  Dieser Baustein ist damit im E-Mail-Template auch als Signatur verfügbar.

1. Basis-BO generiert aus Regelbaustein:

2. Basis-BO generiert aus Unterformular:

**Beschreibung**

---

Text (0) Verfügbar für (1)

Hier können ebenfalls Datenfelder verwendet werden, wenn sie in dem Basis-BO im eingesetzten Template bzw. unter 1. oder 2. definiertem BO existieren. Endet ein Datenfeld auf "\*", dann wird der Baustein nur dann geschrieben, wenn das gekennzeichnete Datenfeld nicht leer ist.

Text (default)

```
Dabei traten folgende Fehler pro Position auf:
<br>
<tr><th class="first">Zeile</th><th>Standort</th><th>Teilprojekt</th><th>Pos</th><th>Erwartet</th><th>Ist</th><th>Änderungsgrund</th><th>Hinweis</th></tr>
##(Importposition (Tabellenzeile))
</table>
```

---

**E-Mail-Template-Baustein**

Name:

Baustein-Typ:  Versand-Bedingung:

Aktiv?

---

Verfügbar für (1)

Wenn die Verfügbarkeit eingeschränkt werden soll, wird dieser Baustein nur dann in einem E-Mail-Template angeboten, wenn das Template auf einer der ang Dies ist für einige Regelbausteine sinnvoll, wenn diese nicht für alle Tabellen sinnvolle Ergebnisse liefern.

Verfügbarkeit einschränken?

Nudet	Tabelle
LMT	Meilenstein Import

## Version 1.1.3

### Beschreibung von Version 1.1.3

Thema	Beschreibung
E-Mail Template	<p>Das BO <i>E-Mail-Template</i> bietet die Möglichkeit, Vorlagen für E-Mails zu erstellen, die dann aus Regeln heraus versendet werden können. Ein Administrator hat damit die Möglichkeit, die von Nuclios versendeten E-Mails selbst zu gestalten und kurzfristig zu ändern. Die Templates bieten hierzu viele Möglichkeiten:</p> <ul style="list-style-type: none"> <li>Der Zugriff auf beliebig weit referenzierte Attribute aus einem frei wählbaren Basis-BO (z.B Auftrag), ähnlich wie bei Nuclios-Formularen</li> <li>Verschiedene Quellen von Adressen (Attribut, Nuclio-parameter, Benutzer, Benutzergruppe, ...)</li> <li>Verschiedene Quellen von Anhängen (Datei, Formular, ...)</li> <li>Verschiedene Sprachen von Betreff, Nachricht und Anhang</li> <li>Textbausteine</li> <li>E-Mail-Empfänger, die E-Mail-Adresse, Anrede und Sprache der E-Mail bestimmen</li> <li>Standardisierte und konfigurierbare Footer und Signaturen</li> <li>Vorschau und Test des Templates</li> </ul> <p>Komplexere E-Mail-Bestandteile, wie z.B. eine Tabelle mit allen Auftragspositionen, müssen durch eine Regel unterstützt werden.</p>

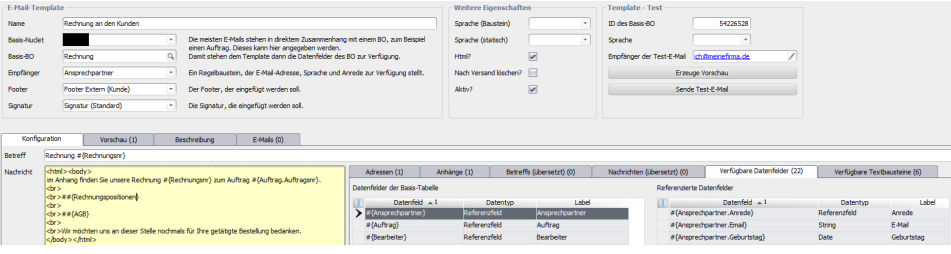
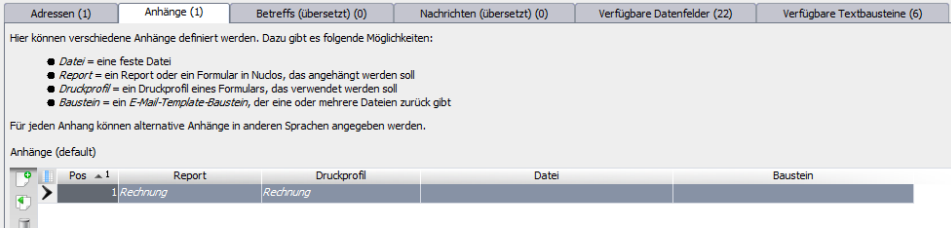
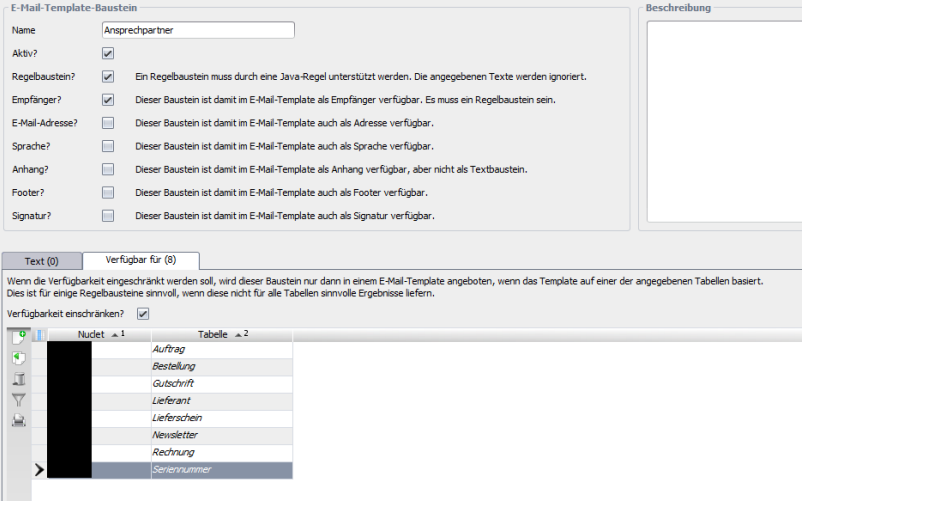
E-Mail Baustein	<p>Können Bausteine definiert werden, die dann in den E-Mail-Templates zur Verfügung stehen. Möglich sind</p> <p>Textbaustein</p> <ul style="list-style-type: none"> <li>• Reine (ggf. mehrsprachige) Standard-Textbausteine (AGB, Versandinfos, Signatur, ...)</li> <li>• Bausteine, die einen Footer oder eine Signatur bereitstellen</li> </ul> <p>Regelbaustein</p> <ul style="list-style-type: none"> <li>• Komplexe Bausteine, die eine Regelunterstützung benötigen (z.B. Tabelle mit Auftragspositionen)</li> <li>• Bausteine, die eine oder mehrere E-Mail-Adressen ermitteln (z.B. für Subscribe-Funktionalitäten)</li> <li>• Bausteine, die eine Sprache definieren</li> </ul> <p>Empfänger</p> <ul style="list-style-type: none"> <li>• Bausteine, die den E-Mail-Empfänger (Adresse, Anrede, Sprache) definieren</li> </ul> <p>Anhang</p> <ul style="list-style-type: none"> <li>• Bausteine, die einen Anhang generieren</li> </ul> <p>Versandbedingung</p> <ul style="list-style-type: none"> <li>• Bausteine, die entscheiden, ob eine E-Mail versendet werden soll oder nicht</li> </ul> <p>Zur Sicherheit lassen sich Bausteine auf bestimmte BO's einschränken, wenn z.B. wenn der Baustein <i>Bearbeiter</i> für die BO's <i>Rechnung</i> und <i>Auftrag</i> existiert, aber nicht für das BO <i>Artikel</i></p>
E-Mail Sprache	Definiert die zur Verfügung stehenden Sprachen
Regelbausteine	<p>Zur Erzeugung einer E-Mail aus einem E-Mail-Template muss eine Instanz einer Klasse mit übergeben werden, die das Interface <i>Regelbausteine</i> implementiert.</p> <p>Das Nucllet bringt eine Implementierung mit, die einige Beispiele enthält. Wenn Sie eigene Regelbausteine benötigen, sollten Sie sich eine eigene Implementierung in Ihrem Nucllet anlegen, damit diese nicht überschrieben wird, wenn das E-Mail-Nucllet mal aktualisiert wird.</p> <p>In der Methode <i>EMailUtils.getRegelBausteine()</i> dann einfach diese Implementierung erzeugen.</p>

### Konfiguration von E-Mail-Templates von Version 1.1.3

Bereich E-Mail-Template		
Name	Pflicht	Eindeutiger Name des Templates
Basis-Nucllet	optional	Das Nucllet aus dem das Basis-BO kommen soll
Basis-BO	optional	Das BO, auf das sich die E-Mail bezieht (z.B. <i>Auftrag</i> )
Empfänger	optional	Ein Regelbaustein, der als <i>Empfänger</i> gekennzeichnet ist und für die die Methode <i>getRecipient()</i> in der Regelbaustein-Implementierung ein Ergebnis mit E-Mail-Adresse, Anrede und ggf. Sprache zurück liefert
Empfänger-BO	optional	Wenn der Empfänger z.B. allgemein als Regelbaustein für das BO <i>Mitarbeiter</i> implementiert wurde, kann hier einfach <i>Mitarbeiter</i> angegeben werden
Empfänger-Rolle	optional	Wenn der <i>Mitarbeiter</i> in verschiedenen Rollen im Basis-BO vorkommt, kann diese hier spezifiziert werden, z.B. <i>Ansprechpartner</i> oder <i>Kundenberater</i>
Footer	optional	Ein als <i>Footer</i> gekennzeichneter Textbaustein
Signatur	optional	Ein als <i>Signatur</i> gekennzeichneter Textbaustein
Bereich Weitere Eigenschaften		
Sprache (Baustein)	optional	Die Sprache, in der die E-Mail versendet werden soll, definiert durch einen als <i>Sprache</i> gekennzeichneten Textbaustein
Sprache (statisch)	optional	Die Sprache, in der die E-Mail versendet werden soll, direkt angegeben
Html?	optional	E-Mail wird im Html-Format versendet (empfohlen)

Nach Versand löschen?	optional	Sensible E-Mails (z.B. mit Passwort-Links o.ä.) können nach dem Versand automatisch aus dem E-Mail-Protokoll gelöscht werden
Aktiv?	optional	Aktiviert oder deaktiviert den Versand der E-Mails
Versandbedingung	optional	Hier kann Regelbaustein ausgewählt werden, der eine Versandbedingung ist, die entscheidet, ob die E-Mail versendet werden soll
<b>Bereich Template-Test</b>		
ID des Basis-BO	optional	Wenn ein Basis-BO angegeben ist, kann hier eine intid angegeben werden, für die die Vorschau erstellt und die Test-E-Mail versendet werden soll
Sprache	optional	Hier kann die Sprache der Vorschau und der Test-E-Mail forciert werden
Empfänger der Test-E-Mail	optional	Wenn kein Empfänger angegeben ist, wird die E-Mail an die im Nuclletparameter <i>Email Umleitung auf</i> hinterlegte Adresse versendet
Erzeuge Vorschau		Erzeugt eine Vorschau im Reiter <i>Vorschau</i>
Sendet Test-E-Mail		Für den finalen Test kann damit eine Test-E-Mail versendet werden
<b>Reiter Konfiguration</b>		
Betreff + Nachricht	optional	Hier können Betreff und Nachricht definiert werden. Dabei können verwendet werden <ul style="list-style-type: none"> <li>• Beliebiger Text</li> <li>• Datenfelder aus dem Basis-BO in der Form z.B. <i>#{Auftragsnr}</i> oder <i>#{Kunde.Kundengruppe.Name}</i>. Verfügbare Datenfelder sind als Hilfe im Reiter <i>Verfügbare Datenfelder</i> zu finden</li> <li>• Textbausteine, die entweder nicht auf ein BO eingeschränkt (<i>##{AGB}</i>) sind oder für das Basis-BO erlaubt sind (<i>##{Ansprechpartner}</i>), als Hilfe zu finden im Reiter <i>Verfügbare Textbausteine</i></li> </ul> Für Betreff und Nachricht können alternative Texte in anderen Sprachen angegeben werden (siehe Reiter <i>Betreffs(übersetzt)</i> und <i>Nachrichten (übersetzt)</i> )
Reiter Adressen	optional	Wenn ein <i>Empfänger</i> angegeben ist, wird die E-Mail an die dort ermittelte E-Mail-Adresse versendet. Zusätzlich oder alternativ können weitere Adressen hinzugefügt werden aus <ul style="list-style-type: none"> <li>• direkt angegebenen Adressen (z.B. CC info@firma.de)</li> <li>• einem Nuclos Benutzer</li> <li>• allen Nuclos Benutzern einer bestimmten Benutzergruppe</li> <li>• einem Attribut des Basis-BO (oder weiter referenzierten Attributen, z.B. die E-Mail-Adresse des Zuständigen des Auftrages)</li> <li>• einem Nucllet Parameter</li> <li>• einem Baustein, der als <i>E-Mail-Adresse</i> gekennzeichnet ist</li> </ul>
Reiter Anhänge	optional	Hier können Anhänge für die E-Mail definiert werden, z.B. <ul style="list-style-type: none"> <li>• ein Report oder Formular</li> <li>• eine im Template hinterlegte Datei</li> <li>• ein Baustein, der als <i>Anhang</i> gekennzeichnet ist und für den die Methode <i>getAttachments()</i> eine Liste von Dateien zurück liefert</li> </ul> Für jeden Anhang können alternative Anhänge in anderen sprachen definiert werden
Reiter Statuswechsel	optional	Hier kann konfiguriert werden, bei welchem Statuswechsel die E-Mail versendet werden soll. Eine entsprechende Regel muss dazu allerdings für jeden möglichen Statuswechsel des betroffenen Statusmodells zugeordnet werden. Hilfreich für diese Regel ist dann die Methode  <i>EMailUtils.sendAllAcceptedEMails(StateChangeContext context, RegelBausteine regelBausteine)</i>
<b>Weitere</b>		
Reiter Beschreibung	optional	Hier können weitere Informationen für das Template hinterlegt werden
Reiter E-Mails		Zeigt die E-Mails an, die bereits aus diesem Template generiert wurden

## Verwendung von E-Mail-Templates von Version 1.1.3

<p><b>Initialisierung</b></p>															
<p>Nach dem Nucletimport muss einmal der Job <i>Initialisiere Nuclet Emailversand</i> ausgeführt werden</p>	<p>Der Job legt u.a. einige Standard-E-Mail-Bausteine an, so dass Sie gleich ein Template bauen können.</p>														
<p><b>E-Mail-Template</b></p>															
<p>Ein E-Mail-Template für eine Rechnung könnte z.B. so aussehen</p>															
<p>Rechnungsformular als Anhang</p>															
<p>Regelbaustein für einen Empfänger</p>															
<p><b>Das Interface Regelbausteine</b></p>															
	<table border="1"> <tr> <td>Text</td> <td>String <b>getValue</b>(String name, Modifiable&lt;Long&gt; bo, EMailTemplateSprache language, TemplateParameters parameters, RuleContext context)</td> </tr> <tr> <td>Anhänge</td> <td>List&lt;NuclonFile&gt; <b>getAttachments</b>(String name, Modifiable&lt;Long&gt; bo, EMailTemplateSprache language, TemplateParameters parameters)</td> </tr> <tr> <td>Empfänger</td> <td>TemplateRecipient <b>getRecipient</b>(String name, Modifiable&lt;Long&gt; bo, EMailTemplateSprache language, TemplateParameters parameters)</td> </tr> <tr> <td>Sprache</td> <td>NuclonLocale <b>getLocale</b>(EMailTemplateSprache language)</td> </tr> <tr> <td>Versandbedingung</td> <td>boolean <b>isSendMail</b>(EMailTemplateBaustein versandbedingung, Modifiable&lt;Long&gt; bo)</td> </tr> <tr> <td>Adresse des Webclients</td> <td>String <b>getWebclientAddress</b>()</td> </tr> <tr> <td>Anrede</td> <td><b>getSalutation</b>(User user, EMailTemplateSprache sprache)</td> </tr> </table>	Text	String <b>getValue</b> (String name, Modifiable<Long> bo, EMailTemplateSprache language, TemplateParameters parameters, RuleContext context)	Anhänge	List<NuclonFile> <b>getAttachments</b> (String name, Modifiable<Long> bo, EMailTemplateSprache language, TemplateParameters parameters)	Empfänger	TemplateRecipient <b>getRecipient</b> (String name, Modifiable<Long> bo, EMailTemplateSprache language, TemplateParameters parameters)	Sprache	NuclonLocale <b>getLocale</b> (EMailTemplateSprache language)	Versandbedingung	boolean <b>isSendMail</b> (EMailTemplateBaustein versandbedingung, Modifiable<Long> bo)	Adresse des Webclients	String <b>getWebclientAddress</b> ()	Anrede	<b>getSalutation</b> (User user, EMailTemplateSprache sprache)
Text	String <b>getValue</b> (String name, Modifiable<Long> bo, EMailTemplateSprache language, TemplateParameters parameters, RuleContext context)														
Anhänge	List<NuclonFile> <b>getAttachments</b> (String name, Modifiable<Long> bo, EMailTemplateSprache language, TemplateParameters parameters)														
Empfänger	TemplateRecipient <b>getRecipient</b> (String name, Modifiable<Long> bo, EMailTemplateSprache language, TemplateParameters parameters)														
Sprache	NuclonLocale <b>getLocale</b> (EMailTemplateSprache language)														
Versandbedingung	boolean <b>isSendMail</b> (EMailTemplateBaustein versandbedingung, Modifiable<Long> bo)														
Adresse des Webclients	String <b>getWebclientAddress</b> ()														
Anrede	<b>getSalutation</b> (User user, EMailTemplateSprache sprache)														
<p><b>Beispiel (Text)</b></p>															

	<pre> public String getValue(final String name, final Modifiable&lt;Long&gt; bo, EmailTemplateSprache language, TemplateParameters parameters, RuleContext context) throws BusinessException {     if (SIGNATUR.equals(name)) {         return EmailHelper.getSignature();     } else if (RECHNUNGSPPOSITIONEN.equals(name)) {         return EmailHelper.getRechnungspositionen(bo, parameters, name);     }      throw new BusinessException("Unbekannter Name für einen Regelbaustein: \" + name + "\"); } </pre>
<b>Beispiel (Empfänger)</b>	
	<pre> @Override public TemplateRecipient getRecipient(String name, Modifiable&lt;Long&gt; bo, EmailTemplateSprache language, TemplateParameters parameters) throws BusinessExc eption {      if (ANSPRECHPARTNER.equals(name)) {         return EmailHelper.getTemplateRecipient(EmailHelper.getAnsprechpartner(bo, parameters, name));     } else if (BEARBEITER.equals(name)) {         return EmailHelper.getTemplateRecipient(EmailHelper.getBearbeiter(bo, name));     }      throw new BusinessException("Unbekannter Name für einen Empfänger: \" + name + "\"); } </pre>
<b>Beispiel (Verwendung in Regeln)</b>	
Einfachster Fall	<pre> EmailTemplate template = EmailUtils.getEmailTemplate(Konstanten.EmailTemplate.RE CHNUNG); if (template.getAktiv()) {     EmailVersand.sendMail(new EmailContext(template, EmailUtils.getRegelBausteine ()), rechnung, context); } </pre>
<b>Erzeugung von Templates (Klasse EmailTemplateKonfiguration)</b>	
<p>Die Klasse EmailTemplateKonfiguration stellt Datenstrukturen und Methoden bereit, um Sprachen, Bausteine und Templates durch Regeln zu erstellen.</p> <p>Dies kann besonders bei der Nucleentwicklung hilfreich sein, um solche Objekte auf anderen Instanzen neu zu erstellen.</p>	
Verfügbare Datenstrukturen	<ul style="list-style-type: none"> <li>• Sprache</li> <li>• Baustein</li> <li>• Template</li> <li>• Restriction</li> <li>• Adresse</li> <li>• Statuswechsel</li> <li>• Anhang</li> <li>• BausteinTranslation</li> <li>• AnhangTranslation</li> <li>• SubjectTranslation</li> <li>• MessageTranslation</li> </ul>



Verfügbare Methoden zur Erstellung	<ul style="list-style-type: none"> <li>• EMailTemplate <b>createTemplate</b>(Template template, final JobContext context)</li> <li>• void <b>createSprache</b>(Sprache sprache, final JobContext context)</li> <li>• void <b>insertBausteinTranslation</b>(EMailTemplateBaustein baustein, BausteinTranslation translation)</li> <li>• void <b>insertBausteinRestriction</b>(EMailTemplateBaustein baustein, Restriction restriction)</li> <li>• EMailTemplateAnhang <b>insertAnhang</b>(EMailTemplate template, Anhang anhang)</li> <li>• void <b>insertStatuswechsel</b>(EMailTemplate template, Statuswechsel statuswechsel)</li> <li>• void <b>insertAnhangTranslation</b>(EMailTemplateAnhang anhang, AnhangTranslation translation)</li> <li>• void <b>insertBetreffTranslation</b>(EMailTemplate template, SubjectTranslation translation)</li> <li>• void <b>insertNachrichtTranslation</b>(EMailTemplate template, MessageTranslation translation)</li> <li>• void <b>insertAdresse</b>(EMailTemplate template, Adresse adresse)</li> <li>• EMailTemplateBaustein <b>createBaustein</b>(Baustein baustein, final JobContext context)</li> </ul>
Beispiel (Konfiguration Baustein)	<pre>public static final EMailTemplateKonfiguration.Baustein FOOTER_INTERN = new EMailTemplateKonfiguration.Baustein() {{     setName(RegelBausteineImpl.Footer.FOOTER_INTERN);     setText("&lt;br&gt;\n&lt;br&gt;Vielen Dank und freundliche Grüße,\n&lt;br&gt;\n&lt;br&gt;##{" + RegelBausteineImpl.Standard.AKTUELLER_BENUTZER + "}");     setFooter(true); }};</pre>
Beispiel (Konfiguration Template)	<pre>public static final EMailTemplateKonfiguration.Template TEMPLATE_PROJEKT = new EMailTemplateKonfiguration.Template() {{     setName("Projekt aktiv");     setNucleus("org.nucleus.firma");     setAktiv(false);     setTable("Projekt");     setTestID(1234567);     setSpracheStatisch(EMailUtils.DEUTSCH);     setSubject("Neues Projekt");     setMessage("##{" + RegelBausteineImpl.Standard.ALLGEMEINE_ANREDE + "}"+ "\n&lt;br&gt;" + "\n&lt;br&gt;das Projekt #{@Nummer} wurde soeben angelegt.");     setFooter(RegelBausteineImpl.Footer.FOOTER_INTERN);     setSignature(RegelBausteineImpl.Signature.SIGNATUR);     addAddress(new EMailTemplateKonfiguration.Adresse() {{         setAttribute("Beauftragtdurch");         setAttributeRef("Emailbenutzer");         setPflichtfeld(true);     }}); }};</pre>
Beispiel Erzeugung	<pre>EMailTemplateKonfiguration.createBaustein(FOOTER_INTERN, jobContext); EMailTemplateKonfiguration.createTemplate(TEMPLATE_PROJEKT, jobContext);</pre>

## Version 1.1.2

### Beschreibung von Version 1.1.2

Thema	Beschreibung
E-Mail Umleitung	<p>Für den Testbetrieb kann die E-Mail Umleitung aktiviert werden. Dabei werden alle ausgehenden E-Mails auf eine festgelegte Adresse umgeleitet.</p> <p>In der E-Mail steht zusätzlich, an wen diese eigentlich gegangen wäre.</p>
E-Mail Protokoll	<p>Alle zu versendenden und alle versendeten E-Mails werden im E-Mail-Protokoll (unter <i>System</i>) gespeichert. Das Speichern der versendeten Emails kann in den Nucleusparametern ausgeschaltet werden.</p>
E-Mail Queue	<p>Für den E-Mail-Versand gibt es zwei verschiedene Möglichkeiten:</p> <p><b>Sofort versenden</b></p> <p>Die E-Mail wird sofort versendet. Ist der Versand von einem Benutzer angestoßen (z.B. durch einen Statuswechsel, der einen E-Mail-Versand auslöst), so muss er warten, bis dieser Vorgang abgeschlossen ist, bevor er weiter arbeiten kann. Kommt es beim E-Mail-Versand zu einem Fehler, so wird die gesamte Aktion (z.B. Statuswechsel) des Benutzers rückgängig gemacht.</p> <p><b>Versand über Email Queue</b></p> <p>Die E-Mail wird erst im E-Mail-Protokoll als nicht versendete E-Mail gespeichert und erst durch einen regelmäßig laufenden Job versendet (<i>Administration &gt; Jobsteuerung &gt; Versende Emails</i>). Dieser muss natürlich aktiviert sein. Der Benutzer wird dadurch bei seiner Bearbeitung nicht unnötig lange aufgehalten und ein fehlerhafter E-Mail-Versand führt nicht zum Abbruch seiner Aktion.</p>

E-Mail Empfang	Der Job <i>E-MailEmpfang</i> erzeugt für neue E-Mails auf dem in den IMAP-Systemparametern angegebenen Server einen Eintrag im E-Mail-Protokoll mit den Flags <i>Versendet = false</i> , <i>Empfangen = true</i> , <i>Verarbeitet = false</i> . Das Flag <i>Verarbeitet</i> kann nach der erfolgreichen Weiterverarbeitung der Mail auf <i>true</i> gesetzt werden.
Verwendung in Regeln	Die Klasse <i>EmailVersand</i> stellt zwei Methoden bereit

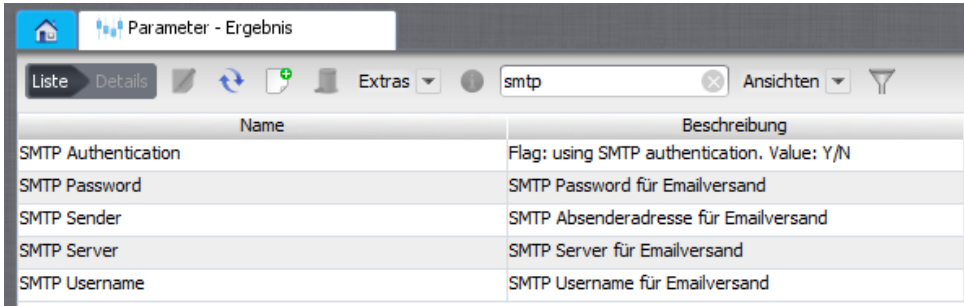
## Verwendung in der Regelprogrammierung von Version 1.1.2

<b>Klasse EMailVersand</b>	
<code>static void sendMail (EmailContext context)</code>	Versendet die im context definierte E-Mail
<code>static void sendMailNow (EmailContext context)</code>	Versendet die im context definierte E-Mail sofort, unabhängig von den Einstellungen in den Nucletparametern
<b>Klasse EMailContext</b>	
<code>void addRecipient(String address, int type)</code>	Fügt einen Empfänger hinzu (type = To, CC oder BCC)
<code>void addRecipient(String address)</code>	Fügt einen Empfänger hinzu (To)
<code>void addRecipients (List&lt;String&gt; lRecipients, int type)</code>	Fügt eine Liste von Empfängern hinzu (type = To, CC oder BCC)
<code>void addRecipients (List&lt;String&gt; lRecipients)</code>	Fügt eine Liste von Empfängern hinzu (To)
<code>void addAttachment (NuclosFile file)</code>	Fügt einen Anhang hinzu
<code>void addAttachments (List&lt;NuclosFile&gt; lAttachments)</code>	Fügt eine Liste von Anhängen hinzu
<code>void setReplyTo(String replyTo)</code>	Definiert (vom Sender abweichende) Antwortadresse
<code>void setMessage(String message)</code>	Definiert die E-Mail-Nachricht
<code>void setSubject(String subject)</code>	Definiert den E-Mail-Betreff
<code>void setReference(String reference)</code>	Definiert eine Referenz. Das kann z.B. eine Auftragsnummer sein. Die Referenz hat keinen Einfluss auf die E-Mail, hilft aber, E-Mails im Protokoll wiederzufinden, die zu einem bestimmten Auftrag gehören.
<code>void resetAttachments()</code>	Leert die Liste der Anhänge
<code>void resetRecipients()</code>	Leert die Liste der Empfänger
<b>Beispiel (E-Mail-Versand)</b>	

	<pre> EmailContext context = new EmailContext(); context.addRecipient(auftrag.getKundeBO.getEmailadresse()); context.addRecipient("chef@firma.de", EmailContext.BCC); context.addAttachment(auftrag.getRechnung()); context.setSubject("Rechnung für Auftrag " + auftrag.getAuftragsnr()); context.setMessage("..."); context.setReplyTo("kundenservice@firma.de"); context.setReference(auftrag.getKundeBO().getKundennr()); EmailVersand.sendMail(context); </pre>
<b>Beispiel (E-Mail-Empfang)</b>	
	<pre> Query&lt;EmailProtokoll&gt; query = QueryProvider.create(EmailProtokoll.class); query.where(EmailProtokoll.Empfangen.eq(Boolean.TRUE)); query.where(EmailProtokoll.Empfangeneemailverarbeitet.eq(Boolean.FALSE)); for (EmailProtokoll email : QueryProvider.execute(query)) {     process(email, context);     email.setEmpfangeneemailverarbeitet(Boolean.TRUE);     email.save(); } </pre>

## Integration von Version 1.1.2

Die Integration nach erfolgreichem Nucletimport (siehe [Nuclet Import](#)) ist wie folgt vorzunehmen.

Schritt	Beschreibung	Screenshot												
1	<p>Konfigurieren Sie die Nuclos Parameter für die Nutzung des MailProviders in Nuclos (unter <i>Administration &gt; Parameter</i>):</p> <ul style="list-style-type: none"> <li>• SMTP Authentication</li> <li>• SMTP Username</li> <li>• SMTP Password</li> <li>• SMTP Server</li> <li>• SMTP Sender</li> </ul> <p>Siehe dazu auch <a href="#">Email (SMTP/IMAP/Pop3)</a>.</p>	 <table border="1"> <thead> <tr> <th>Name</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>SMTP Authentication</td> <td>Flag: using SMTP authentication. Value: Y/N</td> </tr> <tr> <td>SMTP Password</td> <td>SMTP Password für Emailversand</td> </tr> <tr> <td>SMTP Sender</td> <td>SMTP Absenderadresse für Emailversand</td> </tr> <tr> <td>SMTP Server</td> <td>SMTP Server für Emailversand</td> </tr> <tr> <td>SMTP Username</td> <td>SMTP Username für Emailversand</td> </tr> </tbody> </table>	Name	Beschreibung	SMTP Authentication	Flag: using SMTP authentication. Value: Y/N	SMTP Password	SMTP Password für Emailversand	SMTP Sender	SMTP Absenderadresse für Emailversand	SMTP Server	SMTP Server für Emailversand	SMTP Username	SMTP Username für Emailversand
Name	Beschreibung													
SMTP Authentication	Flag: using SMTP authentication. Value: Y/N													
SMTP Password	SMTP Password für Emailversand													
SMTP Sender	SMTP Absenderadresse für Emailversand													
SMTP Server	SMTP Server für Emailversand													
SMTP Username	SMTP Username für Emailversand													

2 Konfigurieren Sie die Nucllet Parameter des *Nuclets Emailversand* (unter *Konfiguration > Nucllet Management*)

Name	Beschreibung
Email Absender	Absender der E-Mails
Email Betreff Prefix	Optionales Prefix für die Betreffzeile aller Nuclos-Mails (OFF = kein Prefix)
Email Nachricht Abschluss	Optionale Abschlussformel der Nachricht (OFF = kein Abschluss)
Email Nachricht Signatur	Optionale Signatur der Nachricht (OFF = keine Signatur)
Email sofort versenden	E-Mails werden sofort versendet und nicht in der E-Mail-Queue zwischen gespeichert (Y/N)
Email Umleitung	Alle E-Mails werden umgeleitet (Y/N)
Email Umleitung auf	Ziel der Umleitung
Versendete Emails löschen	Erfolgreich versendete E-Mails aus dem Email-Protokoll löschen
Empfangene Emails löschen	Empfangene E-Mails werden vom Mail-Server gelöscht

Parameter (8)	Extensions (0)	Release Notes (0)	Abhängigkeiten (0)
- Nucllet Parameter			
	Name ▲ 1	Wert	
	Email Absender	<a href="mailto:nuclos@novabit.de">nuclos@novabit.de</a>	
	Email Betreff Prefix	[Nuclos Test]	
	Email Nachricht Abschluss	   Hier Abschlussformel einfügen	
	Email Nachricht Signatur	  -- Hier Signatur einfügen	
	Email sofort versenden	N	
	Email Umleitung	Y	
	Email Umleitung auf	<a href="mailto:nuclos@novabit.de">nuclos@novabit.de</a>	
	Versendete Emails löschen	N	

## Version 1.1.0

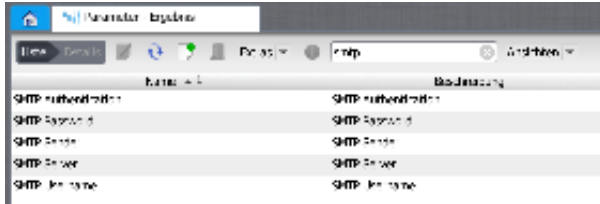
### Integration / Beschreibung von Version 1.1.0

Die Integration nach erfolgreichem Nuclletimport (siehe [Nucllet Import](#)) ist wie folgt vorzunehmen.

Schritt	Beschreibung
---------	--------------

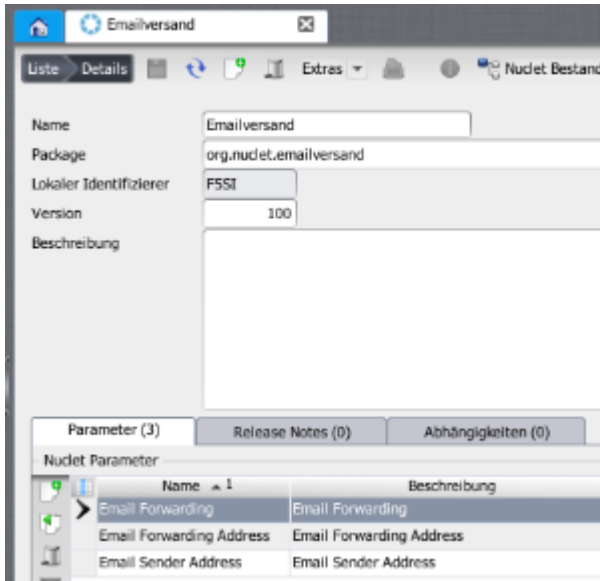
1

Konfigurieren Sie die, für die Nutzung des MailProviders in Nuclis erforderlichen, SMTP-Systemparameter (mindestens SMTP Authentication, SMTP Username, SMTP Password, SMTP Server, SMTP Sender). Siehe dazu auch [Email \(SMTP/IMAP/Pop3\)](#).



2

Konfigurieren Sie die durch Nuclis zu setzende From/ReplyTo-Adresse im Nuclisparameter "Email Sender Address". Aktivieren oder deaktivieren Sie desweiteren wie gewünscht die Umleitung in den Nuclisparametern "Email Forwarding" ("Y" oder "N") und "Email Forwarding Address". Ist die Umleitung aktiviert, werden alle E-Mails, die über dieses Nuclis versandt werden, nicht an den eigentlichen Empfänger gesandt, sondern an die in "Email Forwarding Address" angegebene Adresse umgeleitet. Auf diese Weise lässt sich die Umsetzung automatisch zu versendender E-Mails in Ihrem eigenen Nuclis einfach testen, bzw. es lässt sich damit ein Testsystem leicht so konfigurieren, dass es keine E-Mails an Ihre Kunden oder Lieferanten sendet.



3

Passen Sie die zu verwendende Signatur in der Klasse EMailSignatur an Ihre Anforderungen an.

4

Nutzen Sie die neu gewonnene Funktionalität durch Versand von E-Mails z.B. an Ihre Kunden und Lieferanten. Dazu stehen Ihnen folgende Methoden zur Verfügung:

```
EmailVersand.sendEmail(String sRecipient, String sSubject, String sMessage, NuclosFile attachment, String lang)
```

```
EmailVersand.sendEmail(List<String> lRecipient, String sSubject, String sMessage, NuclosFile attachment, String lang)
```

```
EmailVersand.sendEmail(List<String> lRecipient, List<String> lRecipientCC, List<String> lRecipientBCC, String sSubject, String sMessage, List<NuclosFile> lAttachment, String lang)
```

Über diese Methoden versandte E-Mails werden unter dem Menüpunkt System | E-Mail-Protokoll abgelegt und archiviert. Denken Sie daran, Ihre Anwender bei Bedarf für diesen Menüpunkt lesend zu berechtigen.

Beispiel:

```
String sSubject = "Ihre Rechnung";
```

```
String sMessage = "Im Anhang erhalten Sie Ihre Rechnung";
```

```
(new EMailVersand()).sendEmail(kunde.getEmailadresse(), sSubject, sMessage, pdfInvoice, "de");
```